Learning How to Vote with Principles: Axiomatic Insights Into the Collective Decisions of Neural Networks

LEVIN HORNISCHER*, Munich Center for Mathematical Philosophy, LMU Munich, Germany
ZOI TERZOPOULOU, GATE, CNRS, Université Jean Monnet, Université Lumiere Lyon 2, France

Can neural networks be applied in voting theory, while satisfying the need for transparency in collective decisions? We propose *axiomatic deep voting*: a framework to build and evaluate neural networks that aggregate preferences, using the well-established axiomatic method of voting theory. Our findings are: (1) Neural networks, despite being highly accurate, often fail to align with the core axioms of voting rules, revealing a disconnect between mimicking outcomes and reasoning. (2) Training with axiom-specific data does not enhance alignment with those axioms. (3) By solely optimizing axiom satisfaction, neural networks can synthesize new voting rules that often surpass and substantially differ from existing ones. This offers insights for both fields: For AI, important concepts like bias and value-alignment are studied in a mathematically rigorous way; for voting theory, new areas of the space of voting rules are explored.

JAIR Associate Editor: Alessandro Farinelli

JAIR Reference Format:
Levin Hornischer and Zoi Terzopoulou. 2025. Learning How to Vote with Principles: Axiomatic Insights Into the Collective Decisions of Neural Networks. *Journal of Artificial Intelligence Research* 83, Article 25 (August 2025), 44 pages. DOI: 10.1613/jair.1.18890

1 Introduction

Artificial intelligence (AI) is increasingly applied in many domains, including not just scientific and technological but also societal problems. This poses a dilemma when it comes to *social choice*, i.e., voting, preference aggregation, and other processes of collective decisions. On the one hand, voting systems should be transparent, but the neural networks on which modern AI is built are notoriously opaque. On the other hand, neural networks could unearth novel and tailor-made collective decision procedures. Already, state-of-the-art techniques for alignment of Large Language Models (LLMs) with human values—like RLHF[1] or DPO [49]—rely on the aggregation of human preferences about the generated outputs to fine-tune LLMs. This triggered recent research in guiding such AI alignment using social choice [13].

In this paper, we study how neural networks aggregate votes and preferences. When they form such collective decisions, do they adhere to the normative principles that social choice theory formulates as axioms? This is fundamental both for a discussion of the dilemma and for using social choice for AI alignment. Moreover, it offers new insights for both AI and voting theory. For AI, this provides a rich testing ground to study pressing machine learning concepts like bias, value-alignment and interpretability in a mathematically rigorous way. For example, a network is not biased towards specific individuals if it aggregates their preferences in accordance with the axiom

*Corresponding Author.
[1]Bai et al., 2022. Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback. arXiv:2204.05862.

Authors' Contact Information: Levin Hornischer, ORCID: 0000-0003-2087-527X, levin.hornischer@lmu.de, Munich Center for Mathematical Philosophy, LMU Munich, Munich, Germany; Zoi Terzopoulou, ORCID: 0000-0001-5289-434X, zoi.terzopoulou@cnrs.fr, GATE, CNRS, Université Jean Monnet, Université Lumiere Lyon 2, Saint-Etienne, France.

DOI: 10.1613/jair.1.18890

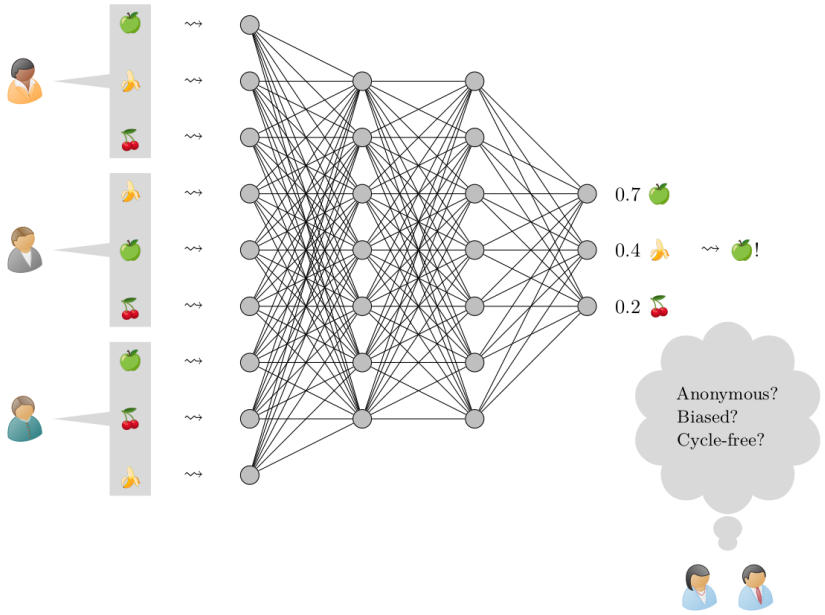

Fig. 1. Can neural networks learn to vote with principles?

of anonymity; the so-called Pareto principle requires the neural network to align with any preference shared among all individuals; and the well-known axiom of independence entails a certain compositional interpretability of the network. For voting theory, axiomatic deep voting provides a new method for the central quest of exploring the space of voting rules.

Social choice. How are individual preferences best turned into a collective decision? This question is studied by *social choice theory* [9, 35], and specifically *voting theory* [55]. A *voting rule* is a function that takes as input a *profile*—i.e., a list of each individual's preferences among a set of alternatives—and produces as output a *collective decision*, i.e., the alternative(s) that the rule takes to be most preferred for the group as a whole (see Section 3 for the formal definitions). The most straightforward rule is *Plurality* (picking the top alternative of most individuals); other classic rules include *Borda* and *Copeland*, while a recent suggestion is *Stable Voting*.

Axiomatic deep voting. To study the collective choices of neural networks, we develop the *axiomatic deep voting* framework (sketched in Figure 1).[2] Deep neural networks are (parametrized) functions that map vectors (typically of a high dimension) to vectors (typically of a low dimension). So, after suitably *encoding* profiles and collective decisions as vectors, neural networks realize voting rules, i.e., functions from profiles to collective decisions. Discovering a voting rule can then be seen as an *optimization problem*: updating the neural network parameters until a given desired property is fulfilled. We *evaluate* a trained neural network in terms of accuracy and axiom satisfaction. While the former is standard in machine learning, the latter is specific to voting theory and its *axiomatic method* [34, 51]. Different axioms describe different desirable properties of voting rules. An example is the already mentioned anonymity axiom which requires that the names of the voters should *not* influence the collective decision.

[2]The source code is available here: https://github.com/LevinHornischer/AxiomaticDeepVoting.

Research questions. With this framework, we investigate the general question of how neural networks aggregate preferences via three more specific questions.

(1) *Correct for the right reasons?* Neural networks can accurately learn standard voting rules, but do they adhere to the normative principles expressed by voting axioms?

We observe eminent violations of the axioms, despite high accuracy in mimicking voting rules. So we focus on teaching neural networks the expert knowledge expressed by axioms. There are two common ways to do this. The first is via *dataset augmentation* [52]:

(2) *Learning principles by example?* Can neural networks be trained to adhere to voting axioms by training with data exemplifying the axioms?

The second way is via *semantic loss functions* [53]. For this, we develop a translation of the axioms into loss functions; so, by optimizing this loss during training, the network increases the corresponding axiom satisfaction. Importantly, though, perfect axiom satisfaction is impossible according to the infamous theorem by Arrow [4]. So we search for the best possible axiom satisfaction:

(3) *Rule synthesis guided by principles?* When neural networks optimize axiom satisfaction, can they develop new voting rules that surpass existing ones?

We compare the discovered rules to a wide range of known voting rules, to test if neural networks can advance the current state of the art in voting theory.

Key findings. In three experiments, we answer these questions in turn. In each, we test three paradigmatic neural network architectures: multi-layer perceptrons, convolutional neural networks, and word embedding based classifiers. We also check a variety of standard distributions of voter preferences. Our three experiments find, respectively:

(1) The employed architectures demonstrate similar behavior both regarding accuracy and axiom satisfaction. Importantly, despite high accuracy, they markedly violate critical axioms like anonymity—yet, the news is not as bad for other axioms.
(2) Data augmentation does not seem to boost the principled learning of neural networks. However, it drastically decreases the amount of required training data.
(3) Neural networks that perform the unsupervised learning task of optimizing axiom satisfaction discover voting rules that are substantially different from existing ones and are comparable—and often better—in axiom satisfaction.

Thus, we fruitfully combine two approaches to studying the space of voting rules: Drawing on machine learning, we use neural networks qua universal function approximators to *explore* that space; and drawing on voting theory, we *evaluate* points in that space—i.e., voting rules—by their axiom satisfaction, thus guiding the exploration.

2 Related Work

We identify three main streams of relevant literature.

2.1 Axiomatic Evaluation of Voting Rules

Social choice theory has extensively quantified the axiom satisfaction of various voting rules, with a significant focus on the concept of *manipulability*, i.e., the propensity of voters to be untruthful in order to sway the outcome in their favor [18, 19, 43]. Numerous studies [20, 39, 45] examine how often voting rules elect the *Condorcet winner* (i.e., the alternative representing a majoritarian consensus) for relatively small elections all having the same probability of materializing (i.e., assuming the Impartial Culture distribution). In line with our findings, the Borda rule is found to elect the Condorcet winner more often than the Plurality rule [45]. When considering

the axiom of independence—the main trigger of Arrow's impossibility theorem—the Borda rule fulfills it more frequently than Copeland, which in turn satisfies it more than Plurality [16]. For the special case of 3 voters and 3 alternatives, an anonymous voting rule satisfies independence between 1.3% and 25.5% of the time [47].

Overall, our work aligns with the traditional concept of evaluating voting rules based on axioms. However, we also consider learning voting rules and not just evaluating them.

2.2 Neural Networks and Voting

The synergy between voting and machine learning has recently garnered more and more attention. [31] use, among others, multi-layer perceptrons (MLPs) on elections of 20 alternatives and 25 voters to predict the winners of different voting rules. The study's primary aim is to identify an effective computational technique on top of the classical ones of the voting literature. The authors find that the Borda rule is predicted by the neural networks with high accuracy (up to 99%), but more complex rules are predicted with lower accuracy (up to 85% for Kemeny and 89% for Dodgson). Burka et al. [10] employ MLPs to investigate the relation between sample size and accuracy when learning different voting rules, including Plurality, Borda, and Copeland. In that work, up to 3000 data points are used based on the Impartial Culture assumption, with at most 5 alternatives and 11 voters. The MLP is found to mimic more closely Borda, no matter on which rule it is trained: e.g., for 3 alternatives and 7 voters, trained on Plurality, the MLP mimics Borda with 95% accuracy and Plurality with 86% accuracy. However, the size of the training data exhibits an impact on the results: e.g., when trained on elections with a Condorcet winner, the MLP mimics more closely Borda in sample-size up to 1000, and Copeland in larger samples. Increasing the size of the MLP by adding layers does not seem important. [2] study more complex neural network architectures (such as Set Transformers and DeepSets), improving the accuracy of MLPs by up to 4% in learning Plurality and Copeland. With sufficiently many data points, those networks are shown to match almost perfectly each voting rule, and to generalize to elections with unseen numbers of voters.

Similarly to all these works, our first experiment considers precisely the problem of using neural networks to learn existing rules from voting theory. However, we systematically study this with axioms: instead of only targeting the right outcomes, we test whether they are obtained via the right principles. In an initial exploration towards the same direction, Armstrong and Larson [3] use a single axiom—prescribing the election of a Condorcet winner when one exists—to train normatively appealing neural networks. This work relies on real data from Canadian federal elections, while ours builds on extensive synthetic data. Additionally, our third experiment illustrates original interactions between sets of different axioms that have not been explored in the literature yet.

After observing a theoretical trade-off between fairness (in particular anonymity variations, demanding that different types of voters be treated equally) and certain notions of economic efficiency (some related to the Condorcet winner), Mohsin et al. [41] train two machine learning models on synthetic data and discover new voting rules that compete well against both Plurality and Borda. Although rule synthesis and axiomatic analysis is not a main focus of that work, the obtained results enforce the idea that machine learning methods can beat existing ones from economic theory when optimized for principled learning. In the slightly different framework of probabilistic voting, MLPs are used to learn voting rules that output distributions over the set of alternatives (rather than the certain winning alternatives), for elections of up to 7 alternatives and 29 voters.[3] It is shown that the discovered rules can lead to novel ones with improved axiomatic properties, after the appropriate embedding of the input and some adjustments of the output. Another work studies the setting of participatory budgeting (PB), where citizens vote about the projects that they would like to implement in their neighbourhoods, while taking into account their respective costs.[4] Set Transformer neural networks trained on PB instances with both real and synthetic data are found to both learn existing voting rules and to discover new ones that satisfy societal

[3]Matone et al., 2024. DeepVoting: Learning Voting Rules with Tailored Embeddings. arXiv:2408.13630.
[4]Fairstein, Vilenchik, and Gal, 2024. Learning Aggregation Rules in Participatory Budgeting: A Data-Driven Approach. arXiv:2412.01864.

objectives (such as fair representation). Overall, our paper adds to this literature by offering a concrete framework to study combinations of basic axioms from the ML perspective, in the most standard voting setting.

Other promising lines of research target learning an abstract voting rule given examples about its choices [48] and designing a voting rule that maximizes some notion of social welfare [2]. Holliday et al. [27] explore the strategic manipulation of voting rules by MLPs of different sizes, generating elections of up to 6 alternatives and 21 voters. They find that sufficiently large MLPs learn to profitably manipulate all examined voting rules only with information about the pairwise majority victories between alternatives. But some rules like Split Cycle seem more resistant than other rules (e.g., Plurality and Borda). Moreover, a systematic study of the connections between social choice and RLHF is conducted by Dai and Fleisig [15], including fundamental axiomatic notions such as the Condorcet winner.

A different approach, rather orthogonal to ours, is to consider AI models as the individuals who vote, instead of using them as the aggregation mechanisms. In this vein, Yang et al. [54] consider a human voting experiment with 180 participants to establish a baseline for human preferences and conducted a corresponding experiment with LLM (e.g., GPT-4) agents. The voting behavior of the networks seems to be affected by the presentation order of the alternatives, as well as the numerical ID assigned to each LLM representing a voter. Some voting rules such as Borda show that LLMs may lead to less diverse collective outcomes. Importantly GPT-4 seems to over-rely on stereotypical demographics of the voters it is supposed to mimic. Similarly, using data from Brazil's 2022 presidential election, Gudiño Rosero et al. [23] tests the accuracy with which LLMs predict an individual's vote. They find that LLMs are more accurate than a naive rule guessing that individuals simply vote for the proposals of the candidate most aligned with their political orientation.

2.3 Social Choice for AI Alignment

A growing research area studies how social choice theory can be used to guide the alignment of modern AI methods with human values and moral judgments. Conitzer et al. [13] highlight a series of technical connections—for example, the alternatives in a voting context could be treated as all possible parameterizations of a network, or as all its possible answers. As an indication, in a popular work about a controversial topic, Noothigattu et al. [44] use data from the online 'moral machine experiment' to build a model of aggregated moral preferences aimed at guiding the decision making of autonomous vehicles. On a more theoretical level, Arrow's theorem can be utilized to prove that there does not exist any AI system that can treat all its users and human supervisors equally.[5] Based on an investment game where participants are asked to manage wealth, Koster et al. [30] find that by optimizing for human preferences, a reinforcement learning model can propose a wealth redistribution mechanism that is supported by the majority of participants. We do not directly engage with the ethical dimension of this research area; still, we participate in the related foundational discussion by studying whether neural networks can learn to vote with principles.

3 Preliminaries on Voting Theory

We work in the standard setting of voting theory, where a finite set N of *voters* have preferences that are linear orders (also called *rankings*) over a finite set A of *alternatives* [55]. Set $m := |A|$ and $n := |N|$. We denote by $\boldsymbol{P} = (P_1, ..., P_n)$ a preference *profile*, i.e., a vector with the preference P_i for every voter $i \in N$. This is illustrated in Figure 2.

For a permutation of the alternatives $\sigma : A \to A$, the ranking $\sigma(P)$ is obtained by applying σ elementwise to the ranking P, and $\sigma(\boldsymbol{P}) = (\sigma(P_1), \ldots, \sigma(P_n))$. For a permutation of the voters $\pi : N \to N$, we define

[5] Mishra, 2023. AI Alignment and Social Choice: Fundamental Limitations and Policy Implications. arXiv:2310.16048.

1	2	3	4	5	6
c	d	d	c	a	d
a	b	b	b	b	a
b	c	a	d	c	b
d	a	c	a	d	c

Fig. 2. A voting profile, with voters $N = \{1, \ldots, 6\}$ and alternatives $A = \{a, b, c, d\}$. Each column depicts the preference of the individual voter; e.g., voter 1 prefers alternative c most, followed by alternative a, etc.

$\pi(\boldsymbol{P}) = (P_{\pi(1)}, ..., P_{\pi(n)})$. A *voting rule* is a function F that determines the winning alternatives for each such profile. Formally, $F : \boldsymbol{P} \mapsto S$, where $\emptyset \neq S \subseteq A$.[6]

3.1 Voting Rules

Voting rules usually fit into one of two categories: scoring rules and tournament solutions. *Scoring rules* assign a score to each alternative depending on its position in the linear preference of each voter and declare as winners those alternatives with the highest score across all voters. The two primary scoring rules are *Plurality* (assigning score 1 to an alternative each time it is ranked first by a voter, and score 0 otherwise) and *Borda* (assigning score $m - 1$ to an alternative ranked first by a voter, score $m - 2$ to an alternative ranked second, and so on, until score 0 is assigned to an alternative ranked last by a voter).[7] Another, less popular scoring rule, is *Anti-Plurality*, which assigns score 0 to an alternative each time it is ranked last by a voter, and score 1 otherwise.

Tournament solutions on the other hand are based on tournaments that capture pairwise comparisons between the alternatives, induced by the voters' preferences. For $x, y \in A$, let $N^{\boldsymbol{P}}_{x>y}$ be the set of voters i in the profile $\boldsymbol{P}$ that consider x better than y in P_i, and $n^{\boldsymbol{P}}_{x>y} := |N^{\boldsymbol{P}}_{x>y}|$. A classical tournament solution is the *Copeland* rule, which selects as winners the alternatives that beat the most other alternatives in a pairwise majority contest: $\text{argmax}_{x \in A} |\{y \in A : n^{\boldsymbol{P}}_{x>y} \geq n^{\boldsymbol{P}}_{y>x}\}|$. In other words, an alternative can be thought to be assigned a Copeland score of 1 for every other alternative to which it is majority preferred, with the winner being the alternative with the highest score overall. Analogously, the *Llull* rule assigns to an alternative a score 1 for every other alternative to which it is majority preferred, and score $1/2$ for every other alternative to which it is majority tied—if there are no majority ties, then the Llull winners coincide with the Copeland winners. The *Top Cycle* rule selects the smallest set of alternatives such that each alternative in the set is majority preferred to each alternative outside the set. In a sense, this set captures the notion of a Condorcet winner when a single such alternative does not exist. For the *Banks* rule, we say that a chain $(x, y, z, \ldots)$ in a tournament is a subset of alternatives that are linearly connected by the majority relation (i.e., x is majority preferred to y, y is majority preferred to z, and so on). Then x is a Banks winner if it is the maximum element of a maximal chain. To define the recently proposed *Stable Voting* rule [26], we first need to describe—the more computationally expensive, and thus left aside in our analysis—*Split Cycle* [25]. The weighted tournament of a profile is a weighted directed graph the nodes of which are alternatives with an edge from x to y of weight $n^{\boldsymbol{P}}_{x>y}$. Suppose that in each cycle of the graph, we simultaneously delete the edges with minimal weight. Then the alternatives with no incoming edges are the winners of Split Cycle. If there is only one Split Cycle winner in a profile $\boldsymbol{P}$, then this also is the winner of Stable Voting; otherwise x is a winner of Stable Voting if for some alternative y it holds that x is a Split Cycle winner with the maximal margin $n^{\boldsymbol{P}}_{x>y}$ such that x is a Stable Voting winner in the profile $\boldsymbol{P}_{-y}$ obtained from $\boldsymbol{P}$ after deleting alternative y.

[6] We use the Python package pref-voting in all our experiments [28].
[7] Plurality and Borda are often contrasted in voting [24, 50].

Other prominent rules that do not fit into the two above categories are *Blacks*, *Baldwin*, and *Weak Nanson*. Black returns the Condorcet winner (i.e., the alternative beating every other alternative in a pairwise strict majority contest) if one exists, otherwise it returns the Borda winners. Weak Nanson (resp., Baldwin) is defined iteratively on voting profiles of various sizes. In each round, all alternatives with below-average (resp., the lowest) Borda score are removed. Whenever all alternatives have the same Borda score, they all win; otherwise the alternative that remains in the last round wins. More voting rules can be defined in iterative terms, based on the idea that less popular alternatives in one round be dropped from all preferences in the next round, until some surviving alternative achieves majority support: *Plurality with Runoff* consists of at most two rounds—if no alternative is ranked on top by a majority of voters in the first round, then the second round selects the majority winner between the two alternatives that achieved the highest Plurality score in the first round; *Instant Runoff* (resp., *Coombs*) is such that in each round the alternatives with the lowest Plurality score (resp., Anti-Plurality score) are eliminated.

A rule different in spirit, the *Uncovered Set*, relies on the idea of undefeated alternatives. Let us say that alternative x left-covers y if all alternatives z that are majority preferred to x are also majority preferred to y. Then, x defeats y if x is majority preferred to y and also left-covers y. The winners of Uncovered Set are the undefeated alternatives. Finally, the *Kemeny-Young* rule is based on a notion of distance between the preferences of the voters: it constructs the linear order that minimises the sum of Kendal Tau distances for all voters' preferences and elects as winner the alternative on the top of that linear order.

Note that all the aforementioned rules are included in our last experiment, which aims at comparing a wide pool of different voting rules that vary in nature. Our other two experiments are focused on the three most standard voting rules, Plurality, Borda, and Copeland, which are the most frequently studied in the literature to date at the intersection of voting and machine learning. For more detailed discussions of the rules, we refer to the introductory chapter of [55] and to the `pref-voting` documentation.

3.2 Axioms

We define axioms as functions that map a voting rule and a preference profile to a value in $\{-1, 1, 0\}$, where 0 means that the axiom is not applicable, -1 means that the axiom is violated, and 1 that it is satisfied. The *satisfaction degree* of a rule with respect to a given axiom is the ratio of the number of sampled profiles in which the axiom is satisfied to the number of sampled profiles in which it is applicable. We focus on axioms that capture basic and diverse normative properties of a voting rule F.

- *Anonymity* is always applicable; it is satisfied in $\boldsymbol{P}$ if for all permutations of voters $\pi : N \to N$, $F(\pi(\boldsymbol{P})) = F(\boldsymbol{P})$. In words, the winners should be invariant under permutations of the voters.
- *Neutrality* is always applicable; it is satisfied in $\boldsymbol{P}$ if for all permutations of alternatives $\sigma : A \to A$, $F(\sigma(\boldsymbol{P})) = \sigma(F(\boldsymbol{P}))$. In words, under permutations of the alternatives, the winners should be permuted respectively.
- *Condorcet principle* is applicable in $\boldsymbol{P}$ if some $x \in A$ is such that $n^{\boldsymbol{P}}_{x>y} > n/2$ for all $y \in A \setminus \{x\}$; it is satisfied if $F(\boldsymbol{P}) = \{x\}$. In words, if a Condorcet winner exists, then it should be the unique winner of the voting rule.
- *Pareto principle* is applicable in $\boldsymbol{P}$ if there exist two alternatives $x, y \in A$ such that $n^{\boldsymbol{P}}_{x>y} = n$; it is satisfied if $y \notin F(\boldsymbol{P})$. In words, if an alternative is considered inferior to a certain other alternative by all voters, then it should not win.
- *Independence* is applicable in $\boldsymbol{P}$ if $F(\boldsymbol{P}) \neq A$; it is satisfied if for all $x \in F(\boldsymbol{P})$, $y \notin F(\boldsymbol{P})$, and $\boldsymbol{P'}$ such that $N^{\boldsymbol{P}}_{x>y} = N^{\boldsymbol{P'}}_{x>y}$, it holds that $y \notin F(\boldsymbol{P'})$. In words, if the relative ranking between a winning alternative and a losing alternative remains the same for all voters, then the losing alternative should not win.

All voting rules defined above satisfy anonymity and neutrality, as well as the Pareto principle, for all preference profiles. They all violate independence for some preference profile. Several of them such as Copeland, Llull, Blacks, Banks, Stable voting, and Weak Nanson always satisfy the Condorcet principle.

3.3 Distributions of Preference Profiles

Specifying the distribution of preference data is essential to studying the voting behavior of a society. Indeed, there is increasing interest within the computational social choice community towards a line of work called 'map of elections' that attempts to systematize simulation experiments relying on synthetic voting data [6, 7]. In this vein, we aim to ensure that our results are independent of the specific choice of the distribution. We thus employ four representative distributions that aim to cover elections of different nature, as explained below. In Section 5.1 we specify the choice of parameters of these distributions for our experiments, which allows them to capture a wide range of realistic scenarios. It is also important to note that all our experimental results are found to be robust across distributions; thus we do not expect that by examining additional distributions we would discover significantly different insights.

Impartial Culture (IC) assumes that all preference profiles have the same probability of appearing. Each preference of a voter in a profile is sampled uniformly at random. The *Mallows* distribution [38] fixes a reference ranking P and assumes that each voter's preference is close to that ranking. Closeness to the reference ranking is defined using the Kendall Tau distance, parameterized by a dispersion parameter $\phi \in (0, 1]$.[8] This distribution reduces to IC when $\phi = 1$ and concentrates all mass on P as ϕ tends to 0.

The IC and Mallows distributions are complementary: IC is simplistic and widely employed in theoretical works on voting rules as discussed earlier in the literature review; it captures an extreme case with no correlation between preferences of voters. Mallows is often employed in numerical studies of voting rules that use artificial data but wish to capture more realistic voting scenarios [12, 33].

The next two distributions also capture more intricate relationships between the preferences in a profile. According to the *2D-Euclidean* distribution, voters and alternatives are distributed randomly in 2-dimensional Euclidean space, and the closer an alternative is to a voter the more the voter prefers that alternative. Finally, the *Urn* distribution [17] generates a profile given a parameter $\alpha \in [0, \infty)$. Voters randomly draw their ranking from an urn. Initially, the urn includes all possible rankings over the alternatives. After a voter randomly draws from the urn, we add to the urn $\alpha n!$ copies of that ranking. When $\alpha = 0$, this reduces to IC.

4 Method

To answer our research questions, we develop the *axiomatic deep voting* framework, visualized in Figure 3. It is built around a neural network, which is a function $f_w : \mathbb{R}^i \to \mathbb{R}^j$ parametrized by weights $w \in \mathbb{R}^k$. We will instantiate this with three different neural network architectures (see Section 4.1). Every profile $\boldsymbol{P}$ is mapped, via an *encoding* function e (see Section 4.2), to a vector $x = e(\boldsymbol{P}) \in \mathbb{R}^i$, for which the neural network produces an output $\hat{y} \in \mathbb{R}^j$.[9] The *decoding* function d (see Section 4.3) turns this output into a winning set $S = d(\hat{y})$. Thus, this setup realizes the voting rule:

$$F_w(\boldsymbol{P}) := d\Big(f_w\big(e(\boldsymbol{P})\big)\Big).$$

[8] The *Kendall Tau distance* between two rankings P and Q over the same set of alternatives is the number of pairs of alternatives (a, b) such that a is preferred over b in P but not so in Q.

[9] For our third architecture, the encoding function is part of the neural network, while for the first two it is independent (see Section 4.2); hence we treat e as a separate entity here.

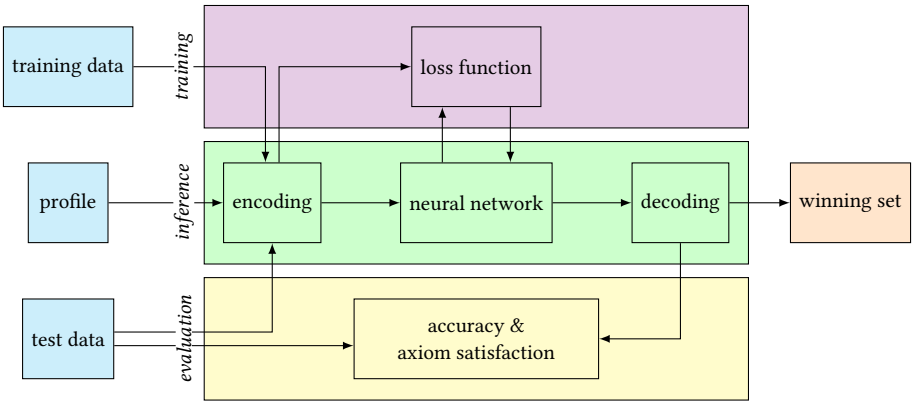

Fig. 3. The axiomatic deep voting architecture.

The network is trained, as usual, using backpropagation with respect to a *loss function* (in Section 4.4), which relies on training data. Finally, we *evaluate* (in Section 4.5) the trained network not only with respect to its accuracy (how well it fits the test dataset), but, crucially, also by how much it satisfies the various voting axioms.

4.1 Architectures

We use three paradigmatic neural network architectures from modern machine learning.

First, *multi-layer perceptrons* (MLPs)—also known as feed-forward neural network—are the classic deep neural net [see, e.g., 21, ch. 6]. They consist of an input layer of neurons, one or more hidden layers, and an output layer.

Second, *convolutional neural networks* (CNNs) are a standard architecture to process grid-like input data such as images [see, e.g., 21, ch. 9], and in our case profiles. Compared to MLPs, they additionally use so-called convolutional layers to capture local, invariant patterns in the input.

Third, we devise an architecture that satisfies the anonymity axiom by design: We view profiles as sentences whose words are the rankings. We use the *word embedding* algorithm *Word2vec* [40] to map each ranking to a high-dimensional embedding vector. These vectors are averaged—hence we get anonymity—and an MLP then classifies this average into a winning set. This combined architecture we call here *word embedding classifiers* (WECs).

4.2 Encoding

To ensure our neural networks learn general patterns, we do not work with a fixed number of voters and alternatives, but only with a maximal number of voters $n_{\max}$ and a maximal number of alternatives $m_{\max}$. So the model should allow as input any profile $\boldsymbol{P}$ over the set of voters $N = \{0, \dots, n-1\}$ with $n \leq n_{\max}$ and set of alternatives $M = \{0, \dots, m-1\}$ with $m \leq m_{\max}$. (For readability, we also write $a, b, c, \dots$ for the alternatives.) We write a_r^s for the r-th most preferred alternative of voter s, so the profile $\boldsymbol{P}$ is represented as the matrix $(a_r^s)_{r,s}$, whose columns are the rankings as in Figure 2. We write $\tilde{\boldsymbol{P}} = (\tilde{a}_r^s)_{r,s}$ for the result of padding the $m \times n$ matrix $\boldsymbol{P}$ with the symbol $\sim$ to the maximal input dimensions $m_{\max} \times n_{\max}$. (So $\tilde{a}_r^s$ is a_r^s if $r \leq m$ and $s \leq n$, and otherwise it is $\sim$.)

How should we encode $\tilde{\boldsymbol{P}}$ so it can be inputted to a neural network? The most straightforward way is to read each alternative $a_r^s \in M$ as the number that it is and the padding symbol $\sim$ as, say, -1. Then the matrix $\tilde{\boldsymbol{P}}$ is regarded as a vector of dimension $m_{\max} n_{\max}$. However, this does not perform well, so, following Anil and Bao [2], we represent an alternative not as a number but as a one-hot vector. For $a \in \{0, \dots, m_{\max} - 1\}$, let $\overline{a}$ be the vector

of length $m_{\max}$ that is 1 at position a and 0 everywhere else. For the padding symbol, let $\overline{\overline{\ }}$ be the vector of length $m_{\max}$ that is 0 everywhere. We write $\overline{\boldsymbol{P}} = (\overline{\tilde{a}_r^s})_{r,s}$.

The *encoding function for MLPs*, e_{MLP}, maps profile $\boldsymbol{P}$ to the vector x obtained by casting the matrix $\overline{\boldsymbol{P}}$ column by column into a flattened vector (of dimension $m_{\max}^2 n_{\max}$). This vector x can then be inputted into the MLP.

The *encoding function for CNNs* regards the matrix $\overline{\boldsymbol{P}}$ as a pixel image: the 'pixel' at position (r, s) has the 'color value' $\overline{\tilde{a}_r^s}$. Thus, e_{CNN} maps profile $\boldsymbol{P}$ to the matrix $\overline{\boldsymbol{P}}$ recast as a tensor with dimensions $(\textit{channel}, \textit{height}, \textit{width}) = (m_{\max}, m_{\max}, n_{\max})$. This tensor can then be inputted into the CNN.

The *encoding function for WECs* regards the profile $\boldsymbol{P} = (P_1, \ldots, P_n)$ as a sentence with words P_i. We train it to embed these words into vectors of a fixed high dimension. Thus, unlike the previous encoding functions, this one is not separate from the neural network but rather forms the first layer of the WEC, with the remaining layers processing the embedding vectors. More precisely, we first pre-train the embeddings as follows. For a given corpus size c, we sample c-many profiles from a given distribution of profiles (e.g., IC) to form our corpus (i.e., a set of sentences). The rankings occurring in the profiles form the vocabulary of this corpus, to which we add the unk token (to later represent *unknown* rankings, i.e., rankings that are not in the vocabulary) and the pad token (to *pad* a profile to length $n_{\max}$). Due to the unk token, this encoding applies to *all* profiles, even if it contains rankings that are not part of the model's vocabulary. Using Word2vec, we train embeddings which represent words in the vocabulary as vectors. When instantiating the WEC architecture, these embeddings form the first layer: it maps the profile $(P_1, \ldots, P_n)$ to the corresponding embedding vectors $(v_1, \ldots, v_n)$. The next layer averages these vectors into a single vector v, followed by several linear layers ending with the output layer.

4.3 Decoding

Given a profile $\boldsymbol{P}$ as input, all neural network architectures produce as output the logits $\hat{y} = (\hat{y}_0, \ldots, \hat{y}_{m_{\max}})$ in $\mathbb{R}^{m_{\max}}$. We apply the sigmoid function sig elementwise to obtain the probability that alternative r is in the winning set.[10] With m the number of alternatives in profile $\boldsymbol{P}$, we define the decoding function

$$d_m(\hat{y}) := \big\{r \in \{0, \ldots, m\} : \text{sig}(\hat{y}_r) > 0.5\big\}.$$[11]

In experiment 3, we will consider further versions of this decoding function (see Section 6.3).

4.4 Loss Functions

Since multiple alternatives can win, we cast the task of finding a voting rule as a *multi-label classification* problem. Each input profile $\boldsymbol{P}$ is associated with m binary labels (where m is the number of alternatives in $\boldsymbol{P}$), and the r-th label is 1 if and only if the r-th alternative is in the winning set associated with $\boldsymbol{P}$. Hence we use *binary cross entropy* as loss function.

A main contribution of this paper is that, for each axiom, we also design a loss function that enforces satisfaction of that axiom. So, for each axiom ax, we define a function $L_{\text{ax}}(f_w, \boldsymbol{P})$ that takes as input the function f_w computed by the neural network with weights w and a profile $\boldsymbol{P}$. It outputs a non-negative real number describing numerically how much the axiom is satisfied: 0 means perfect axiom satisfaction, while higher numbers mean worse axiom satisfaction. We now define these loss functions, which we will use later.

[10] We use 'sig' instead of 'σ' to denote the sigmoid function, in order to not confuse it with the previous use of 'σ' for permutations of alternatives.

[11] A priori, it can happen that the neural network does not assign any winner, in contrast to our definition of a voting rule. We check (and train) that this happens, if at all, only with a negligible probability.

Anonymity. Given the network f_w and profile $\boldsymbol{P}$, uniformly sample N-many permutations $\pi_1, \ldots, \pi_N$ of the set of voters of $\boldsymbol{P}$ and define

$$L_A(f_w, \boldsymbol{P}) := \frac{1}{N} \sum_{r=1}^{N} \mathrm{KL}\Big(f_w\big(e(\boldsymbol{P})\big), f_w\big(e(\pi_r(\boldsymbol{P}))\big)\Big),$$

where KL is Kullback–Leibler divergence.[12]

Condorcet. If $\boldsymbol{P}$ has no Condorcet winner, $L_C(f_w, \boldsymbol{P}) := 0$, and otherwise, if that Condorcet winner is alternative a, define (recall $\overline{a}$ is the one-hot vector for alternative a)

$$L_C(f_w, \boldsymbol{P}) := \mathrm{KL}\big(f_w(e(\boldsymbol{P})), \overline{a}\big).$$

Pareto. We define (recall that $n^{\boldsymbol{P}}_{a>b} = n$ means that all voters in $\boldsymbol{P}$ rank a above b)

$$L_P(f_w, \boldsymbol{P}) := \sum_{a,b \text{ with } n^{\boldsymbol{P}}_{a>b}=n} \mathrm{sig}\Big(f_w(e(\boldsymbol{P}))_b\Big).$$

Independence. Define $L_I(f_w, \boldsymbol{P}) := 0$ if $\boldsymbol{P}$ does not have at least two alternatives. Otherwise, randomly sample N-many pairs (a_r, b_r) of distinct alternatives in $\boldsymbol{P}$ and randomly sample, for each ranking P_k of $\boldsymbol{P} = (P_1, \ldots, P_n)$, a shuffling P'_k of P_k in which, however, the order of a_r and b_r is the same as in P_k, and set $\boldsymbol{P}_r := (P'_1, \ldots, P'_n)$. Write $\hat{y} := f_w(e(\boldsymbol{P}))$ and $\hat{y}^r := f_w(e(\boldsymbol{P}_r))$, and define

$$L_I(f_w, \boldsymbol{P}) := \sum_{r=1}^{N} \mathrm{KL}\Big(\big(\hat{y}_{a_r}\hat{y}_{b_r}\big), \big(\hat{y}^r_{a_r}\hat{y}^r_{b_r}\big)\Big).$$

No winner. Recall that voting rules are required to output at least one winner. This is usually not called an axiom, and we did not hard-code this into our architectures. So we also want to optimize our neural networks to align with this requirement. Hence we define the 'no winner' loss as follows. Writing $\hat{y} = f_w(e(\boldsymbol{P}))$, we want that at least one of the numbers in $p := \big(\mathrm{sig}(\hat{y}_1), \ldots, \sigma(\hat{y}_m)\big)$ is above 0.5, i.e., the maximum norm $\|p\|_\infty$ should be above 0.5. Hence the more it is below that, the worse the loss:

$$L_{NW}(f_w, \boldsymbol{P}) := \max\big(0.5 - \|p\|_\infty, 0\big).[13]$$

4.5 Evaluation Metrics

We have two ways of evaluating the model: accuracy and axioms. First, we calculate the accuracy of the trained neural network on a given test set in two ways: *Identity* (or *hard*) *accuracy* is the percentage of pairs $(\boldsymbol{P}, S)$ in the test set for which $F_w(\boldsymbol{P}) = S$. *Subset* (or *soft*) *accuracy* is defined in the same way but replacing the identity with $F_w(\boldsymbol{P}) \subseteq S$. Second, we calculate the satisfaction degrees for the various axioms of the voting rule F_w that the trained neural network realizes (see Section 5.3 for details).

5 Experimental Setup

We describe all details for designing and evaluating our experiments.

[12]Though, in principle, other distance/similarity functions can be considered.

[13]To see almost-everywhere differentiability of the loss functions, use the distributivity of the differential operator over sums, the chain rule, and the almost-everywhere differentiability of the involved functions (KL, sig, max, $\|\cdot\|_\infty$).

5.1 Voting-Theoretic Parameters

We work with all four profile distributions and with $n_{\max} = 77$ and $m_{\max} = 7$ in the first experiment and $n_{\max} = 55$ and $m_{\max} = 5$ in the other experiments. The first experiment does not show a qualitative difference between these settings, but the latter is computationally more efficient.

For all experiments, we use the Mallows distribution with a parameter rel-ϕ (randomly generated) that, together with the number of alternatives, determines the value of the dispersion parameter ϕ. According to [6] and [7], this methodology generates data that more closely resemble those of real elections. We use the Urn-R distribution [6], where, for each generated profile, α is chosen according to a Gamma distribution with shape parameter $k = 0.8$ and scale parameter $\theta = 1$. The other distributions do not need further parameters.

5.2 Synthetic Data Generation

We can sample profiles in a controlled and realistic manner and produce their corresponding winning sets with existing voting rules (see Section 3). So we generate synthetic data: Given a profile distribution μ and a voting rule F, we randomly pick integers $n \in [1, n_{\max}]$ and $m \in [1, m_{\max}]$ and μ-sample a profile $\boldsymbol{P}$ with n voters and m alternatives and compute $S = F(\boldsymbol{P})$. Thus, we generate a dataset $D = \{(\boldsymbol{P}_1, S_1), \ldots, (\boldsymbol{P}_k, S_k)\}$.

5.3 Evaluating Axiom Satisfaction

To evaluate the axiom satisfaction of a voting rule (be it realized by a neural network or an existing one), we sample 400 profiles on which the axioms are applicable. We use the same profile distribution μ as was used for training the neural network, and we again randomly choose integers $n \in [1, n_{\max}]$ and $m \in [1, m_{\max}]$ before μ-sampling a profile with n voters and m alternatives. To compute whether an axiom is satisfied for a profile, the axioms of anonymity, neutrality, and independence require sampling of permutations. We sample, per profile, 50, 50, and $w(m-w)256$ permutations, respectively (where w is the number of winners according to the rule on the profile, and hence $m - w$ is the number of losers).[14]

5.4 Hyperparameters

All models use ReLU as the activation function. Our MLP has four hidden layers with 128 neurons each, like those of [2]. The CNN has two convolution layers with kernel size $(5, 1)$ and $(1, 5)$, respectively (and 32 or 64 channels), followed by three linear layers with 128 neurons. Thus, the first kernel can pick up local patterns in the rankings of the voters, while the second kernel can pick up local patterns among the i-th preferred alternatives of the voters. (Appendix A.2 establishes the optimality of this choice when compared to other kernel sizes and additional pre-processing.) The WEC has the word embedding layer, then the averaging layer, and then three linear layers with 128 neurons. For pre-training the word embedding layer with word2vec, we use a corpus size of 10^5, an embedding dimension of 200, and a window size of 7.[15] The corpus size is chosen large enough so that no occurrences of the unk token are observed in 1,000 sampled profiles.

This results in the following numbers of parameters in the setting $n_{\max} = 77$ and $m_{\max} = 7$: 500,487 (MLP), 1,834,439 (CNN), and 1,226,143 (WEC). In the setting $n_{\max} = 55$ and $m_{\max} = 5$ this reduces to: 193,285 (MLP), 232,165 (CNN), and 45,585 (WEC). Thus, the models have roughly comparable capacities. Section A in the Appendix motivates these choices via hyperparameter tuning.

[14] Independence considers more 'degrees of freedom', so we take more samples. Specifically, we go through all pairs (x, y) where x is a winner and y a loser, and we sample, for each voter, 4 alternative rankings that, however, have the same relative order of x and y as the actual ranking submitted by that voter, and then build $4^4 = 256$ profiles out of these alternative rankings and check that y still does not win.
[15] That is in the setting $n_{\max} = 77$ and $m_{\max} = 7$. When $n_{\max} = 55$ and $m_{\max} = 5$, we reduce this to a corpus size of 2×10^4, an embedding dimension of 100, and a window size of 5.

For training, we use the *AdamW* algorithm [37]. We use a batch size of 200. Since we have synthetic data, we do not use epochs and hence only specify the number of gradient steps. In experiments 1, 2, and 3, these are 15,000, 5,000, and 15,000, respectively. Similar to Anil and Bao [2], we use as a learning rate scheduler cosine annealing with warm restarts [36]. All results are reported for one fixed seed. (In the Appendix, Table 3 performs cross validation and Tables 5 and 8 report averaged results across different seeds.) All experiments were run on a laptop without GPU.

6 Results and Analysis

Within our axiomatic deep voting framework, we answer our three research questions: (1) Are preferences-aggregating neural networks correct for the right reasons? No. (2) Can they learn voting-theoretic principles by example? No. (3) Can they synthesize new rules guided by the principles? Yes.

6.1 Experiment 1: Correct for the Right Reasons?

Recent work in computer science has studied the capabilities of neural networks to learn voting rules [2, 10], but without asking whether "the system performs well for the right reasons" [5, p. 5192]. Here we use voting-theoretic axioms to shed light on the learning behavior of neural networks, specifically aiming to distinguish solely accurate versus principled learning.

Design. We train each of the three neural network architectures (MLP, CNN, and WEC) on data from each one of the three basic voting rules (Plurality, Borda, and Copeland) using four different sampling distributions (IC, Urn, Mallows, and Euclidean) with the parameters mentioned in Section 5.1. We report the results as *relative* accuracy and axiom satisfaction, i.e.,

$$\langle \text{relative evaluation} \rangle = \langle \text{rule evaluation} \rangle - \langle \text{model evaluation} \rangle.$$

For example, if the model has 95% accuracy, then, since the rule has 100% accuracy, there is a relative accuracy *loss* of $100\% - 95\% = 5\%$. If the model has 35% satisfaction of the independence axiom and the rule only 30%, then the relative independence satisfaction is $30\% - 35\% = -5\%$, so there is a relative independence *gain* of 5%.

Results. The relative accuracy and axiom satisfaction when sampling with the IC distribution are given in Figure 4. (Section B in the Appendix shows similar results for the other distributions.) The three architectures do not differ much in accuracy. The best accuracy is achieved for the simple Plurality rule, while the complex Copeland rule decreases accuracy.

Notably, across all voting rules, architectures, and distributions, we see large losses in neutrality despite only low losses in accuracy (e.g., 4.6% relative identity-accuracy loss but 19.5% relative neutrality loss for the WEC architecture when trained on the Plurality rule). Large anonymity losses are also observed under the MLP and CNN architectures (the WEC is anonymous by design). This is particularly noteworthy since anonymity and neutrality are 100% satisfied by the given voting rules. The MLP and CNN models regularly show larger neutrality losses than anonymity losses (with the models trained on Plurality demonstrating the smallest such difference).

Regarding the other axioms, all models adhere perfectly to Pareto, in accordance with the voting rules on which they are trained. The MLP and WEC models trained on Plurality seem to exhibit relative Condorcet gains, but Condorcet losses are found for the CNN model. Along a similar line, the MLP model trained on Borda obtains relative Condorcet gains, but this is not the case for the CNN and WEC models. Since Copeland always satisfies the Condorcet principle, there is a relative Condorcet loss for all models—yet, it is rather small. The MLP and WEC models trained on Plurality and Borda, as well as the CNN model trained on Plurality, satisfy independence to a similar degree as the rules do on which they are trained. All models trained on Copeland exhibit relative independence gains, and the same holds for the CNN model trained on Borda.

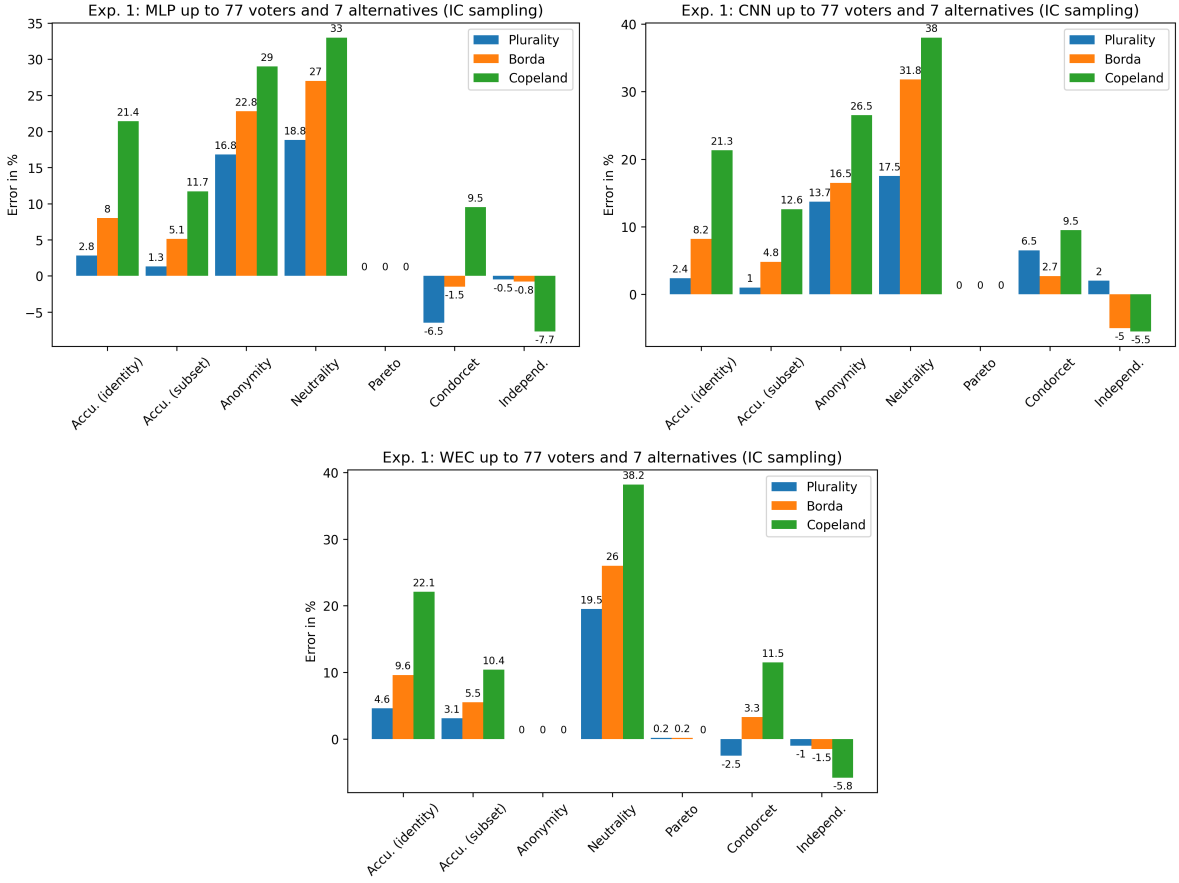

Fig. 4. Training the three architectures (MLP, CNN, and WEC) on data from Plurality, Borda, and Copeland (the three bars in each plot) with IC samples and comparing the errors in both accuracy and axiom satisfaction. The error describes the rule's evaluation minus the model's evaluation. Intuitively, 2.8% accuracy error means 2.8% loss in accuracy: the rule by definition is correct so it has 100% accuracy, but the model obtains only 97.2% accuracy; similarly, −6.5% Condorcet error means 6.5% gain in accuracy: the rule has 73.25% Condorcet satisfaction, but the model obtains 79.75% Condorcet satisfaction.

Discussion. Regarding the learnability of different voting rules, the simplicity of Plurality is probably the reason behind the high accuracy with which all models learn it. However, this simplicity also renders Plurality problematic in other contexts [32].

The models take a stance on the well-documented tension between anonymity and neutrality.[16] They tend to favor outcomes that align more closely with the former than with the latter. This inclination exposes an inherent bias within neural networks when navigating fundamental democratic axioms.

[16]No voting rule that always elects a single winner can simultaneously be anonymous and neutral.

As the architectures are not invariant to permutations of the input data, the severe violations of neutrality (and of anonymity for MLPs and CNNs) are not *a priori* surprising. However, these violations persist even for high accuracy with respect to rules that are perfectly neutral and anonymous.[17]

Overall, our experiment on accurate versus principled learning highlights the importance of the *reasons* behind automated decision-making. Outcomes that mimic well-defined voting rules are arguably still unreliable, since they do not come with a guarantee of respecting the principles on which those rules are built.

6.2 Experiment 2: Learning Principles by Example?

Can we teach neural networks voting-theoretic principles, beyond merely presenting data from various voting rules? A natural approach to integrate expert knowledge in neural networks is data augmentation. In the voting context, this was proposed by Xia [52] but has not been tested in practice, to the best of our knowledge. We focus on the anonymity and neutrality axioms, since they were violated most, and we also test the effects of data augmentation on the model's accuracy.

Asking if data augmentation helps can be understood in two ways: First, does training with augmented data increase axiom satisfaction without diminishing accuracy? Second, if so, does it do this better than just training with sampled data points? A 'yes' to the first question improves data efficiency: we get at least as good a model even when only part of the data is 'real' and the rest is augmented. A 'yes' to the second question means that we can actually improve our model's performance in the preceding experiment.

Design. We test data augmentation in two versions. In the *first version*, we form an initial dataset (i.e., pairs of a sampled profile with corresponding winning set) and we train an architecture on this dataset. Then we continue training it on augmented data points, i.e., data points obtained from the initial data points by renaming alternatives ('neutrality variations') or by renaming voters ('anonymity variations'). For comparison, we make a copy of our model after the initial training and continue training the copy with the same number, but sampled data points (rather than augmented data points). If the model trained on augmented data improves axiom satisfaction without worsening accuracy, we get a 'yes' to the first question. If it also is better than the copied model, we also get a 'yes' to the second question. Otherwise, any improvement only comes from a mere increase in the quantity of data points and not from their quality.

A potential issue of this version is that, during the continued training with augmented data, the model does not see any further sampled data and hence might lose in accuracy. The *second version* fixes this issue by making sure that each training batch consists of p percent sampled data points with the remaining data points being neutrality variations (resp., anonymity variations) of those sampled data points. For different choices of p, we then test the models' achieved axiom satisfaction and accuracy. We get a 'yes' to the first (resp., second) question if axiom satisfaction and accuracy are not worse (resp., better) for lower values of p when compared to $p = 100\%$ (i.e., only sampled data).

Results. Results for IC-sampling and neutrality augmentation are exhibited in Figure 5. (Appendix C presents results also for other distributions and anonymity augmentation.) Overall, we find a 'yes' to the first question but a 'no' to the second: training with augmented data does not hurt accuracy, but it does not reliably improve axiom

[17] Actually, whether this is surprising depends on which of the following two intuitions one has. The first intuition regards these results as surprisingly bad: Given the high accuracy, we may expect that the neural network should have 'gotten the idea' of the voting rule, and hence of its anonymity and neutrality, so the amount of violations is surprising. The second intuition regards the results as surprisingly good: For anonymity and, respectively, neutrality to be satisfied on a given profile, we require the neural network to output the correct answer on 50 permutations of the profile, while accuracy requires being correct only on that very profile, so it is not surprising that the neural network struggles more with anonymity and neutrality. But whether surprising or not, the results stay the same: for the specified percentages of considered profiles, we have violations in anonymity and neutrality, i.e., we do *not* have high certainty (at least one failure in 50 checks) that the neural network outputs the desired answer under permutations.

satisfaction. In the first version of the experiment depicted at the top row and the left plot of the second row in Figure 5, augmented data does not seem to be advantageous for neutrality relatively to sampled data (on the contrary, it often seems harmful, e.g., when applying CNN or WEC); in the second version of the experiment depicted at the bottom row and the right plot of the second row in Figure 5, the ratio p between sampled and augmented data does not seem to correlate with neutrality satisfaction either.

In more detail, regarding the first version, the conclusion of our experiment 1 is again apparent: even when the models excel in accuracy, their satisfaction of neutrality and anonymity remains consistently below perfect—this does not change when considering augmented data. For the CNN and the WEC, neutrality satisfaction with augmented data is (almost) always below the corresponding one with sampled data, while for the MLP it is mixed. The accuracy of the models is generally not hurt by augmented data. Indeed, it does not seem to vary much between sampled and augmented data. Though, in some cases, data augmentation is detrimental to accuracy: e.g., the identity accuracy of the CNN after 1000 gradient steps (on top of 500 steps of pretraining) is up to 10% lower with augmented data than with sampled data.

Regarding the second version, when $p < 10\%$, i.e., with almost only augmented data, both accuracy and neutrality satisfaction are unsatisfactory, so data augmentation only becomes relevant for $p \geq 10\%$. Here accuracy is stable: it does not vary by more than 5%. In some cases, neutrality is equally stable: for the CNN on all rules and the MLP on Borda (certainly for $p \geq 25\%$, with slightly worse neutrality satisfaction for smaller p). In the remaining non-stable cases, the best neutrality satisfaction is achieved for $p = 100\%$, i.e., without augmented data—with only negligible exceptions.[18] Thus, neither in the stable nor the unstable cases can we see reliable comparative improvements in neutrality satisfaction with more neutrality augmented data.

Discussion. Learning voting-theoretic principles by examples—augmented to the training data—does not seem to work for neural networks: Comparatively more neutrality-augmented or anonymity-augmented data does not necessarily lead to higher neutrality or anonymity satisfaction. However, an advantage of data augmentation is a drastic increase in data efficiency when we only aim for accuracy. For instance, sampling only 10% of the total data set (and using neutrality augmented data for the remaining 90%) does not substantially decrease the MLP's or WEC's accuracy in comparison to sampling the whole data set. This is crucial if we use real and not sampled election data, where having access to a vast amount of data points is practically impossible. Even when more data is needed to increase the accuracy of network, we could build an appropriate data set based on a limited amount of real data points and then augment it via the neutrality axiom. This gives us an answer to the two questions raised earlier on: 'Yes', training with additional, augmented data points can increase axiom satisfaction, but 'no', not better than just training with equally many sampled data points.

6.3 Experiment 3: Rule Synthesis Guided by Principles?

We saw that neural networks, when trained on data from established voting rules, struggle to vote with principles. This raises the question: can we directly train neural networks to form principled collective decisions, without relying on any pre-existing voting rules? This will be limited by Arrow's Impossibility Theorem [4]: a voting rule cannot simultaneously satisfy anonymity, Pareto, and independence. Neural network-based approaches also face this impossibility. However, how close can we get to full axiom satisfaction? We design an optimization task, using custom loss functions, to guide neural networks in learning novel and principled voting rules.

[18]The only two exceptions are the CNN on Plurality (where neutrality is most satisfied at $p = 75\%$ but to a very similar degree as for $p = 100\%$) and the CNN on Copeland (where neutrality is minimized at $p = 25\%$). Moreover, CNN on Borda and MLP on Copeland have a local—albeit not global—minimum at $p = 25\%$. Thus, while there might be some special cases where neutrality is improved in the highly augmented scenario, this is not enough to consider data augmentation as a successful strategy to improve neutrality satisfaction (which is what we are concerned with here).

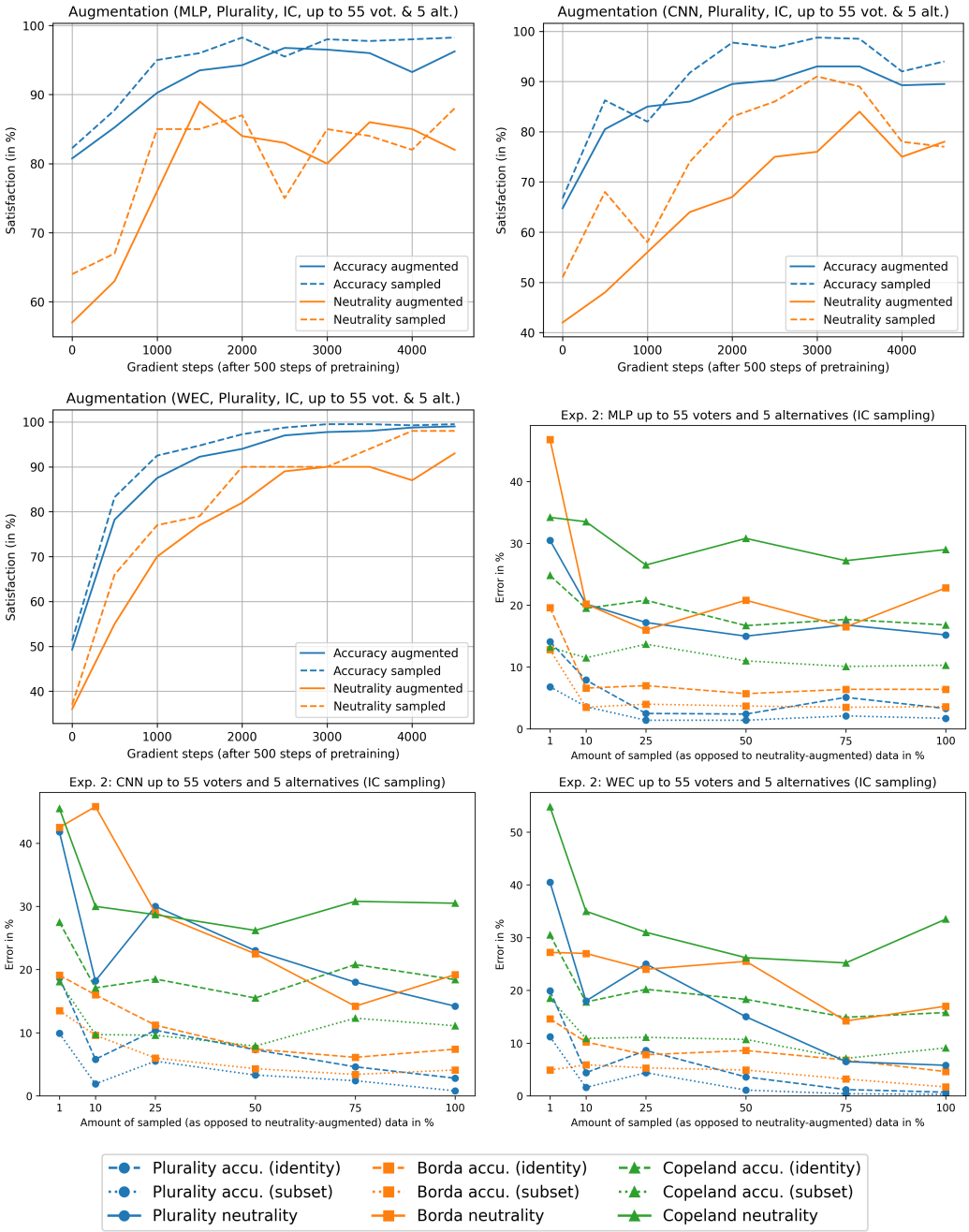

Fig. 5. Top row and second row on the left: The first version. Pretraining on IC-sampled data from Plurality, continuing with neutrality variations and comparing this with identity accuracy to sampled data. Rest: The second version.

Design. We train each one of the three neural network architectures (MLP, CNN, and WEC) on the loss functions defined in Section 4.4 which represent the axioms anonymity, neutrality, Condorcet, Pareto, and independence. Since neural networks could attempt to vacuously satisfy the axioms by proposing no winner, we also consider the "No-winner" loss function, which demands the winning sets to be nonempty. Moreover, by Arrow's Theorem, the axioms cannot be jointly satisfied and will, hence, negatively influence each other. So optimizing for all axioms is not necessarily the best. We pick, for each architecture, a set O of objectives that we optimize for. For WEC, we choose: no winner, Condorcet, and Pareto. (Appendix D.5 establishes in an ablation study the optimality of this choice.) For MLP and CNN, we add: anonymity and independence. Then the optimization problem is:

$$\operatorname*{argmin}_{w} \sum_{O \in \mathcal{O}} \mathop{\mathbb{E}}_{\mathbf{P} \sim D} \left[L_O(f_w, \mathbf{P}) \right],$$

where the loss functions L_O are described in Section 4.4 and D is the chosen distribution of profiles $\mathbf{P}$. Note that, unlike the previous experiments, this is an unsupervised learning task.

In order to have an architecture that is also neutral by design (not just anonymous by design like the WEC), we design a further decoding function in addition to the one used so far (Section 4.3). This *neutrality-averaged* decoding works as follows [cf. 10]. Given an input profile, we first generate all alternative-permuted versions of the profile, then compute the logits-predictions of the model on each of those permuted profiles (in one batch), next de-permute the predictions again and average all of them, and finally we turn those average logits into a winning set with the decoding function used so far.

Thus, WEC with neutrality-averaged decoding is anonymous and neutral by design. For the other architectures, we also test a decoding method that is *neutrality-and-anonymity-averaged*. For that, given an input profile, we first randomly generate 12 alternative-permuted versions of it, and, for each of those, we also randomly generate 10 voter-permuted versions and, as before, compute the averaged logits and from those the winning set. The numbers are explained as follows: Neutrality-averaging requires, with at most 5 alternatives, considering at most $5! = 120$ permutations; hence neutrality-and-anonymity-averaging also considers $12 \times 10 = 120$ permutations (checking all $55! \approx 10^{73}$ voter permutations would be infeasible).

Results. Table 1 shows the axiom satisfaction of different neural networks (bottom) and, for comparison, of several known rules from voting theory (top), all using IC sampling. (Appendix D shows the results for the other distributions. See also Figure 16 in the Appendix for the evolution of the loss during optimization.)

The best neural-network based rule in terms of axiom satisfaction is always the neutrality-averaged WEC, with close contestants the neutrality-averaged CNN and MLP. The neutrality-averaged WEC outperforms the classic Plurality, Borda, and Copeland rules in every single axiom, except for a slight loss on Condorcet. Even when we consider more modern rules in voting theory, the neutrality-averaged WEC is competitive: the existing rule with highest axiom satisfaction is Stable Voting and its edge is marginal, with its average axiom satisfaction being less than 1% higher than that of the neutrality-averaged WEC.[19] In fact, when averaging five runs of checking axiom satisfaction (which always involves some stochasticity), the neutrality-averaged WEC even comes out better than the rules: see Table 5 in the Appendix. (This is also true for Euclidean but not for Mallows and Urn.) In any case, the neutrality-averaged WEC has a comparably good axiom satisfaction as the best voting rules known today.

In addition to examining axiom satisfaction, we should also consider how often the examined rules produce the same outcomes: because similar axiom satisfaction does not imply similarity of outcomes.[20] Table 2 describes similarity in outcome compared to the five closest rules, using IC sampling (again, see Appendix D for the other distributions). In particular, we see that the rule discovered by the neutrality-averaged WEC model is substantially

[19] Table 4 in the Appendix suggests that more gradient steps do not further improve the results.

[20] For example, the Blacks and Weak Nanson rules are close in average axiom satisfaction (less than 1% difference), but Table 2 shows that more than 8% of the time they propose a different set of winners.

Table 1. Axiom satisfaction of different rules (top part of the table) and models (bottom part of the table), for IC sampling. Rounded to one decimal. The names of the models are explained as follows: The letters after the architecture type indicate how the voting rule is computed from the model: p–plain (i.e., no averaging), n–neutrality-averaged, na–neutrality-and-anonymity-averaged. The letters in the brackets indicate which axioms the model optimized for during training: NW–No winner, A–Anonymity, C–Condorcet, P–Pareto, I–Independence. All models have been trained for 15k gradient steps with batch size 200.

	Anon.	Neut.	Condorcet	Pareto	Indep.	Average
Plurality	100	100	80.2	100	28.5	81.8
Borda	100	100	95.5	100	37.2	86.5
Anti-Plurality	100	100	74.2	100	24.8	79.8
Copeland	100	100	100	100	28.0	85.6
Llull	100	100	100	100	26.8	85.4
Uncovered Set	100	100	100	100	27.8	85.5
Top Cycle	100	100	100	100	29.0	85.8
Banks	100	100	100	100	27.8	85.5
Stable Voting	100	100	100	100	43.0	88.6
Blacks	100	100	100	100	35.2	87.1
Instant Runoff TB	100	100	94.8	100	28.2	84.6
PluralityWRunoff PUT	100	100	95.0	100	25.5	84.1
Coombs	100	100	96.2	100	34.5	86.2
Baldwin	100	100	100	100	39.2	87.9
Weak Nanson	100	100	100	100	40.0	88.0
Kemeny-Young	100	100	100	100	39.2	87.9
MLP p (NW, A, C, P, I)	77.8	75.8	92.5	100	39.5	77.1
MLP n (NW, A, C, P, I)	89.2	100	95.0	100	42.2	85.3
MLP na (NW, A, C, P, I)	89.8	86.8	95.5	100	36.5	81.7
CNN p (NW, A, C, P, I)	85.2	67.2	92.0	100	39.5	76.8
CNN n (NW, A, C, P, I)	92.2	100	94.5	100	40.0	85.4
CNN na (NW, A, C, P, I)	86.0	86.5	94.8	100	34.0	80.2
WEC p (NW, C, P)	100	72.5	94.2	100	41.8	81.7
WEC n (NW, C, P)	100	100	96.8	100	41.2	87.6

different from the existing voting rules: it proposes different outcomes than each one of them, according to identity accuracy, at least 9.3% of the time (resp., 10.6% for Mallows, 11.1% for Urn, and 7.8% for Euclidean, see Tables 7, 10, and 13 in the Appendix). In comparison, Stable Voting, which was found best in Table 1, disagrees with Borda and Copeland 8.9% of the time and with Weak Nanson and Blacks only 6.6% of the time. Thus, the discovered rule not only is competitive in axiom satisfaction, it also is novel, i.e., substantially different from existing voting rules.

Moving from hard to soft accuracy, the neutrality-averaged WEC produces winning sets that are a subset (resp., superset) of those of existing rules at least 95.4% (resp., 95.3%) of the time for IC, and similarly for other distributions. This means that even if the results of the model differ from those of existing voting rules, very frequently they do so by only excluding or by only adding certain winning alternatives. This is not unique to our ML model—it is also the case between known voting rules that exhibit high axiomatic satisfaction: for example, the outcome of Stable Voting is a subset (resp., superset) of the outcome of Weak Nanson 97.9% (resp., 94.35%) of the time.

To illustrate the difference between the discovered and the existing rules, Figure 6 shows an example of a profile where the winning set provided by the neutrality-averaged WEC is different to all the winning sets provided by

Table 2. Similarities between the rules. Computed on 10,000 IC-sampled profiles. For example, in the top table ('identity accuracy'), the entry 90.5 in row 'WEC n' and column 'Stable Voting' means that, in 90.5% of the sampled profiles, Stable Voting outputs the identical winning set as the neutrality-averaged WEC. Hence the values below the diagonal are symmetric and thus omitted. In the bottom table ('subset accuracy'), the entries show when the rule in the row outputs a winning set that is a subset of the rule in the column. So the entry 92.7 in row 'WEC n' and column 'Stable Voting' means that, in 92.7% of the sampled profiles, the winning set outputted by the neutrality-averaged WEC is a subset of the winning set outputted by Stable Voting. This table is not symmetric, because the entry 95.4 in row 'Stable Voting' and column 'WEC n' means that, in 95.4% of the sampled profiles, the winning set outputted by Stable Voting is a subset of the model's winning set (equivalently, the model's winning set is a superset of the rule's winning set).

Identity accuracy	WEC n	Blacks	Stable Voting	Borda	Weak Nanson	Copeland
WEC n	100	91	90.5	89.5	88.4	87.7
Blacks		100	95.71	95.13	91.26	90.57
Stable Voting			100	90.84	93.5	91.67
Borda				100	86.39	85.7
Weak Nanson					100	92.43
Copeland						100

Subset accuracy	WEC n	Blacks	Stable Voting	Borda	Weak Nanson	Copeland
WEC n	100	93.6	92.7	93.9	93	95.3
Blacks	95.6	100	96.53	97.3	95.57	98.2
Stable Voting	95.4	97.21	100	94.51	97.9	99.59
Borda	93.8	95.13	91.66	100	90.7	93.33
Weak Nanson	92	92.69	94.35	89.99	100	97.71
Copeland	90.4	91.29	91.73	88.59	94.15	100

1	2	3	4	5	6	7	8
a	*e*	*d*	*a*	*e*	*b*	*e*	*a*
b	*b*	*b*	*c*	*b*	*a*	*a*	*b*
e	*d*	*c*	*b*	*c*	*e*	*c*	*d*
d	*a*	*e*	*e*	*a*	*c*	*d*	*e*
c	*c*	*a*	*d*	*d*	*d*	*b*	*c*

$\{a\}$ neutrality-averaged WEC, with sigmoids (rounded) a:.51, b:.49, c:.31, d:.32, e:.43
$\{b\}$ Blacks, Stable Voting, Borda, Weak Nanson, Copeland
$\{a, e\}$ Plurality, PluralityWRunoff PUT
$\{e\}$ Instant Runoff TB, Anti-Plurality
$\{a, b\}$ Llull, Uncovered Set, Banks, Coombs, Baldwin, and Kemeny-Young
$\{a, b, e\}$ Top Cycle

Fig. 6. Profile where the 'WEC n' model disagrees with existing voting rules. The winning sets for each rule are mentioned below the table.

the considered existing voting rules. The choice of the WEC also has intuitive plausibility: it chooses alternative a which, among the eight voters, is three times the most preferred option and two times the second-most preferred option. The sigmoids indicate that alternative b was also a close competitor for the winning set, and would indeed win under many of the known rules from voting theory.

Discussion. The reason why the WEC outperforms the other two architectures is that, because it is anonymous by design, it is enough to use the neutrality-averaged decoding to get a model that is anonymous and neutral. Since the MLP and CNN are not anonymous, they need neutrality-and-anonymity-averaged decoding to become anonymous and neutral by design. This, however, needs infeasibly many permutations, so it can only be approximated via sampling permutations. Here, however, the tension between the axioms of anonymity and neutrality resurfaces: sampled neutrality-and-anonymity-averaging can result in negative interference with the other axioms yielding worse performance than just neutrality-averaging (e.g., the CNN rules in Table 1). Mere neutrality-averaging also influences the satisfaction of the other axioms, but in this case not in a negative way.[21]

Moreover, for the WEC just three optimization objectives were enough to obtain the above competitive results. Since the MLP and CNN are not anonymous by design, they needed to optimize for anonymity and independence as well. The WEC interestingly had enough *implicit* inductive bias toward satisfying independence—again highlighting non-trivial interference of the axioms and the network architecture.

The neural networks beat the classic voting rules in terms of axiom satisfaction while being comparable to the best voting rules known today. This may be taken to suggest that existing rules may already be close to optimal axiom satisfaction. In other words, they are in the (approximate) *Pareto front* of axiom satisfaction. At the same time, even if the novel rules derived from axiom optimization inherit the opacity of neural networks, they assure high adherence to key normative principles in collective decisions. Since these newly discovered rules were substantially different from existing rules, they extend the boundaries of what is so far explored in voting theory.

7 Discussion

With our axiomatic deep voting framework, we investigated the space of all voting rules by fruitfully combing voting theory and machine learning. The neural network explores the space and the voting-theoretic axioms evaluate the network, thus guiding the exploration. The universal approximation theorems [14, 29] ensure that the neural networks are dense in the space of all voting rules, so all areas of that space can be explored with axiomatic deep voting. Arrow's Impossibility Theorem [4] establishes insurmountable divisions of that space: e.g., the area of rules satisfying anonymity and Pareto does not intersect the area of rules satisfying independence.

The importance of our results for AI is twofold. First, the axiomatic evaluation offers another cautionary tale that accuracy is not everything: Neural networks can have high accuracy (descriptively good) without following the right reasons (normatively bad). Second, this changes, however, when we move from the supervised setting of learning rules from examples to the unsupervised setting of directly optimizing axiom satisfaction. We were able to do this by translating the voting-theoretic axioms into corresponding loss functions. Having a way to optimize the axioms is important because the axioms can be seen as *mathematical formalizations* of important normative notions in modern machine learning. For example:

- *Bias*: anonymity says that the neural network is not biased towards particular individuals.
- *Fairness*: neutrality demands that the neural network treats all alternatives equally.
- *Value-alignment*: the Pareto principle requires that if all individuals value one alternative more than another, then the neural network aligns with this; and similarly for the Condorcet principle.
- *Interpretability*: independence provides a sense of 'compositionality' when interpreting the network—to understand its choice for two given alternatives, we can ignore all other alternatives.

Hence, our axiomatic optimization provides a way of improving the neural network—in a mathematically precise sense—regarding bias, fairness, value-alignment, and interpretability.

Moreover, qua interdisciplinary project, our results are also relevant for voting theory. Axiomatic deep voting offers a new tool for the field's central goal of exploring the space of all voting rules. While existing voting rules

[21]We did not use neutrality-averaging in the previous experiments because it would not directly correspond to the binary cross entropy loss and the interference with the other axioms blurs the axiomatic evaluation of the neural network.

are crafted by human insight, we could find—in a completely automated process—novel voting rules that are comparable in terms of axiom satisfaction to the best rules known today. This provides a promising starting point for an analytic exploration of new axiom-optimal voting rules and the influence the axioms exert on each other.

Limitations. We tested a wide range of standard neural network architectures. However, future work could also investigate further architectures like Set Transformers, Graph Isomorphism Networks, or Deep Sets (which were used by Anil and Bao [2]) and, more generally, the transformer architecture (as a refinement of our word embedding architecture). We also covered the most important voting-theoretic axioms, but yet more can be considered, e.g., monotonicitiy and transitivity (the latter then requires architectures that are not transitive by design). Finally, the large number of permutations causes a high statistical variance in testing the satisfaction of the independence axiom.

Future work. First, more options in generating the dataset can be explored. For example, we can consider the *extrapolation* task in which the model has to find a general rule after only observing the rule on a small part of the input space, namely the profiles where some given voting rules agree or satisfy a given axiom. Or we can consider the *interpolation* task in which the model sees data of different rules and has to find a compromise between their outputs.

Second, we can implement further social choice theory frameworks. Since we already output logits corresponding to the alternatives, we could, instead of winning sets, also consider preference rankings or welfare functions. It would also be interesting to consider judgment aggregation, which includes reasoning about logical implications between the alternatives.

Third, it seems promising to bridge notions of explainability in voting theory [8, 11, 42] and notions of explainability in AI [1]. In particular, is it possible to extract out a symbolic representation (e.g., in logic programming) of the rule that the model learned?

Fourth, studying the voting-theoretic concept of manipulatability via neural networks [27] can be further connected to machine learning notions like *adversarial attacks* [22] or *performativity* [46].

Fifth, from the point of view of *geometric deep learning*,[22] axioms represent *symmetries* that the neural networks should learn. For example, anonymity says that the neural network should be invariant under the group action of the voter-permutation group on profiles; and neutrality says that the neural network should be equivariant under the group action of the alternative-permutation group on profiles and winning sets, respectively. It seems worth exploring this connection to geometric deep learning.

8 Conclusion

We introduced the axiomatic deep voting framework to study how neural networks aggregate preferences. We found that neural networks *do not* learn to vote with principles, despite achieving high accuracy, when trained on data from existing voting rules—even when augmented with axiom-specific data. However, they *do* learn to vote with principles when they directly optimize for axiom satisfaction, which we achieved by translating axioms into custom loss functions. The axiomatic deep voting framework promises fruitful further investigation both in voting theory (new ways of exploring the space of voting rules) and AI (a mathematically precise testing ground for normative notions like bias and value-alignment).

Acknowledgments

For very helpful comments and discussions, we would like to thank Ben Armstrong, Balder ten Cate, Timo Freiesleben, Ronald de Haan, Thomas Icard, Alina Leidinger, Christian List, Ignacio Ojea, as well as the audiences at the 'Workshop on Learning and Logic 2025' at the University of Amsterdam and the 'Social Choice for AI

[22] See the work of Bronstein et al., 2021. Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges. arXiv:2104.13478.

Ethics and Safety 2025' workshop at AAMAS in Detroit. We are especially grateful to the JAIR editor Alessandro Farinelli and two anonymous reviewers.

References

[1] A. Adadi and M. Berrada. 2018. Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). *IEEE Access* 6 (2018), 52138–52160.

[2] Cem Anil and Xuchan Bao. 2021. Learning to Elect. *Advances in Neural Information Processing Systems* 34 (2021), 8006–8017.

[3] Ben Armstrong and Kate Larson. 2019. Machine Learning to Strengthen Democracy. In *NeurIPS Joint Workshop on AI for Social Good*.

[4] Kenneth J Arrow. 1951. *Social Choice and Individual Values*. John Wiley & Sons.

[5] Emily M. Bender and Alexander Koller. 2020. Climbing towards NLU: On Meaning, Form, and Understanding in the Age of Data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 5185–5198.

[6] Niclas Boehmer, Robert Bredereck, Piotr Faliszewski, Rolf Niedermeier, and Stanisław Szufa. 2021. Putting a Compass on the Map of Elections. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI)*.

[7] Niclas Boehmer, Piotr Faliszewski, and Sonja Kraiczy. 2023. Properties of the Mallows Model Depending on the Number of Alternatives: A Warning for an Experimentalist. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*.

[8] Arthur Boixel, Ulle Endriss, and Ronald de Haan. 2022. A Calculus for Computing Structured Justifications for Election Outcomes. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI)*.

[9] Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia (Eds.). 2016. *Handbook of Computational Social Choice*. Cambridge University Press.

[10] Dávid Burka, Clemens Puppe, László Szepesváry, and Attila Tasnádi. 2022. Voting: A Machine Learning Approach. *European Journal of Operational Research* 299, 3 (2022), 1003–1017.

[11] Olivier Cailloux and Ulle Endriss. 2016. Arguing about Voting Rules. In *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

[12] Ioannis Caragiannis and Evi Micha. 2017. Learning a Ground Truth Ranking Using Noisy Approval Votes. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*.

[13] Vincent Conitzer, Rachel Freedman, Jobst Heitzig, Wesley Holliday, Bob Jacobs, Nathan Lambert, Milan Mossé, Eric Pacuit, Stuart Russell, Hailey Schoelkopf, Emanuel Tewolde, and William Zwicker. 2024. Position: Social Choice Should Guide AI Alignment in Dealing with Diverse Human Feedback. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*.

[14] G. Cybenko. 1989. Approximations by Superpositions of Sigmoidal Functions. *Mathematics of Control, Signals, and Systems* 2, 4 (1989), 303–314.

[15] Jessica Dai and Eve Fleisig. 2024. Mapping Social Choice Theory to RLHF. In *ICLR Workshop on Reliable and Responsible Foundation Models*.

[16] Keith L Dougherty and Jac C Heckelman. 2020. The Probability of Violating Arrow's Conditions. *European Journal of Political Economy* 65 (2020), 101936.

[17] Florian Eggenberger and George Pólya. 1923. Über die Statistik Verketteter Vorgänge. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik* 3, 4 (1923), 279–289.

[18] Pierre Favardin and Dominique Lepelley. 2006. Some Further Results on the Manipulability of Social Choice Rules. *Social Choice and Welfare* (2006), 485–509.

[19] Pierre Favardin, Dominique Lepelley, and Jérôme Serais. 2002. Borda rule, Copeland Method and Strategic Manipulation. *Review of Economic Design* 7 (2002), 213–228.

[20] Peter C Fishburn and William V Gehrlein. 1982. Majority Efficiencies for Simple Voting Procedures: Summary and Interpretation. *Theory and Decision* 14, 2 (1982), 141–153.

[21] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. The MIT Press, Cambridge, Massachusetts.

[22] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In *International Conference on Learning Representations (ICLR)*.

[23] Jairo Gudiño Rosero, Umberto Grandi, and César A. Hidalgo. 2024. Large Language Models (LLMs) as Agents for Augmented Democracy. *Philosophical Transactions A* 382, 2285 (2024), 20240100.

[24] Aleksandar Hatzivelkos. 2018. Borda and Plurality Comparison with Regard to Compromise as a Sorites Paradox. *Interdisciplinary Description of Complex Systems: INDECS* 16, 3B (2018), 465–484.

[25] Wesley Holliday and Eric Pacuit. 2023. Split Cycle: A New Condorcet-Consistent Voting Method Independent of Clones and Immune to Spoilers. *Public Choice* 197, 1 (2023), 1–62.

[26] Wesley Holliday and Eric Pacuit. 2023. Stable Voting. *Constitutional Political Economy* 34, 3 (2023), 421–433.

[27] Wesley H. Holliday, Alexander Kristoffersen, and Eric Pacuit. 2025. Learning to Manipulate under Limited Information. In *Proceedings of the 39th AAAI Conference on Artificial Intelligence (AAAI)*.

[28] Wesley H Holliday and Eric Pacuit. 2025. pref_voting: The Preferential Voting Tools package for Python. *Journal of Open Source Software* 10, 105 (2025), 7020.
[29] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. 1989. Multilayer Feedforward Networks are Universal Approximators. *Neural Networks* 2, 5 (1989), 359–366.
[30] Raphael Koster, Jan Balaguer, Andrea Tacchetti, Ari Weinstein, Tina Zhu, Oliver Hauser, Duncan Williams, Lucy Campbell-Gillingham, Phoebe Thacker, and Matthew Botvinick. 2022. Human-centred Mechanism Design with Democratic AI. *Nature Human Behaviour* 6, 10 (2022), 1398–1407.
[31] Hanna Kujawska, Marija Slavkovik, and Jan-Joachim Rückmann. 2020. Predicting the Winners of Borda, Kemeny, and Dodgson Elections with Supervised Machine Learning. In *EUMAS Multi-Agent Systems and Agreement Technologies Workshop*. 440–458.
[32] Jean-François Laslier. 2011. And the Loser is… Plurality Voting. *Electoral Systems* (2011), 327–351.
[33] David Lee, Ashish Goel, Tanja Aitamurto, and Helene Landemore. 2014. Crowdsourcing for Participatory Democracies: Efficient Elicitation of Social Choice Functions. In *Proceedings of the 2nd AAAI Conference on Human Computation and Crowdsourcing*.
[34] Christian List. 2011. The Logical Space of Democracy. *Philosophy & Public Affairs* 39, 3 (2011), 262–297.
[35] Christian List. 2022. Social Choice Theory. In *The Stanford Encyclopedia of Philosophy* (Winter 2022 ed.), Edward N. Zalta and Uri Nodelman (Eds.). Metaphysics Research Lab, Stanford University.
[36] Ilya Loshchilov and Frank Hutter. 2017. SGDR: Stochastic Gradient Descent with Warm Restarts. In *International Conference on Learning Representations (ICLR)*.
[37] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations (ICLR)*.
[38] Colin L Mallows. 1957. Non-Null Ranking Models. *Biometrika* 44, 1/2 (1957), 114–130.
[39] Samuel Merrill. 1984. A Comparison of Efficiency of Multicandidate Electoral Systems. *American Journal of Political Science* (1984), 23–48.
[40] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. *Advances in Neural Information Processing Systems* 26 (2013).
[41] Farhad Mohsin, Ao Liu, Pin-Yu Chen, Francesca Rossi, and Lirong Xia. 2022. Learning to Design Fair and Private Voting Rules. *Journal of Artificial Intelligence Research* 75 (2022), 1139–1176.
[42] Oliviero Nardi, Arthur Boixel, and Ulle Endriss. 2022. A Graph-Based Algorithm for the Automated Justification of Collective Decisions. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.
[43] Shmuel Nitzan. 1985. The Vulnerability of Point-Voting Schemes to Preference Variation and Strategic Manipulation. *Public choice* 47 (1985), 349–370.
[44] Ritesh Noothigattu, Snehalkumar Gaikwad, Edmond Awad, Sohan Dsouza, Iyad Rahwan, Pradeep Ravikumar, and Ariel Procaccia. 2018. A Voting-Based System for Ethical Decision Making. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*.
[45] Hannu Nurmi. 1988. Discrepancies in the Outcomes Resulting from Different Voting Schemes. *Theory and Decision* 25 (1988), 193–208.
[46] Juan Perdomo, Tijana Zrnic, Celestine Mendler-Dünner, and Moritz Hardt. 2020. Performative Prediction. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*.
[47] Robert C Powers. 2007. The Number of Times an Anonymous Rule Violates Independence in the 3× 3 Case. *Social Choice and Welfare* 28, 3 (2007), 363–373.
[48] Ariel D Procaccia, Aviv Zohar, Yoni Peleg, and Jeffrey S Rosenschein. 2009. The Learnability of Voting Rules. *Artificial Intelligence* 173, 12-13 (2009), 1133–1149.
[49] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. *Advances in Neural Information Processing Systems* 36 (2024).
[50] Zoi Terzopoulou. 2023. Voting with Limited Energy: A Study of Plurality and Borda. In *Proceedings of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.
[51] William Thomson. 2001. On the Axiomatic Method and its Recent Applications to Game Theory and Resource Allocation. *Social Choice and Welfare* 18, 2 (2001), 327–386.
[52] Lirong Xia. 2013. Designing Social Choice Mechanisms Using Machine Learning. In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.
[53] Jingyi Xu, Zilu Zhang, Tal Friedman, Yitao Liang, and Guy Broeck. 2018. A Semantic Loss Function for Deep Learning with Symbolic Knowledge. In *International conference on machine learning (ICML)*.
[54] Joshua Yang, Damian Dailisan, Marcin Korecki, Carina Hausladen, and Dirk Helbing. 2024. LLM Voting: Human Choices and AI Collective Decision-Making. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*.
[55] William S Zwicker. 2016. Introduction to the Theory of Voting. In *Handbook of Computational Social Choice*, Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia (Eds.). Cambridge University Press.

A Hyperparameter Tuning

In this section, we investigate different hyperparameter choices for our models to motivate the choices we make in the main text.

A.1 Model Sizes

First, regarding the MLPs, Figure 7 tests the performance of different sizes. As mentioned in Section 5.4, we use the same sizes for our MLPs as Anil and Bao [2]: four hidden layers with 128 neurons each. To test this choice, we compare it to a smaller MLP with only two hidden layers with 128 neurons each, and to a larger MLP with six hidden layers with 128 neurons each plus layer norm. Figure 7 shows that all three MLPs achieve very similar accuracy. The larger MLP learns a bit more quickly than the other two, but it also has a higher variance in achieved accuracy. Hence the larger MLP does not yield a performance improvement. The results suggest that a smaller MLP might work, too, but, for continuity with the literature on the topic, when then choose their model sizes.

Second, regarding the CNNs and the WECs, the choice for the MLP size also dictates their sizes: in order to have a comparable capacity, they should have a roughly similar number of parameters. Indeed, with our choices of numbers and kinds of layers for the CNN and the WEC, we get, as described in Section 5.4, models that are roughly comparable in size.

A.2 CNN Kernels and Pre-processing

Figure 8 investigates different choices for the hyperparameters of the CNN architecture. In Section 5.4, we described our choice: the two convolution layers have kernel sizes $(5, 1)$ and $(1, 5)$, respectively. Thus, the first kernel can pick up local patterns in the rankings of the voters, while the second kernel can pick up local patterns among the i-th preferred alternatives of the voters.

In image processing, 'quadratic' kernel sizes—e.g., $(3, 3)$—are more common, to pick up correlations of pixels with their surrounding pixels. In the voting setting, at least conceptually speaking, a quadratic surrounding does not make too much sense: Why should there be important 'diagonal' correlations, say between the i-th

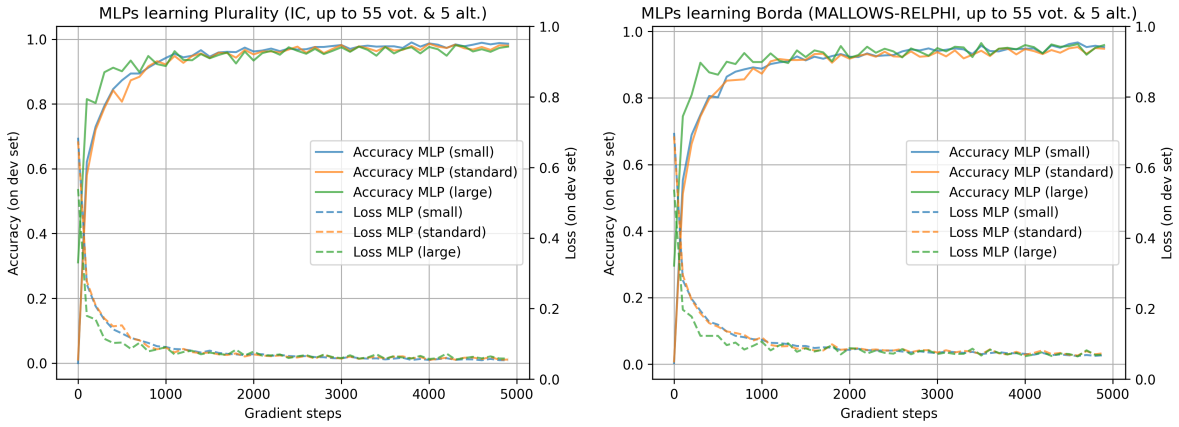

Fig. 7. Different sized MLPs and their learning performance. The 'small' MLP has two hidden layers with 128 neurons each, the 'standard' MLP has four hidden layers with 128 neurons each (our and the literature's choice), and the 'large' MLP has six hidden layers with 128 neurons each plus layer norm.

preferred alternative of voter k and the $i+2$-th preferred alternative of voter $k+3$, especially if the voters should be permutable? On the contrary, vertical and horizontal correlations are important: Vertically, the first kernel captures patterns of correlation between a given alternative in a voter's ranking and more or less preferred alternatives in that ranking; horizontally, the second kernel captures patterns of correlation between the i-th preferred alternative of a voter and the i-th preferred alternative of other voters.

Figure 8 shows that our choice of kernel size (top left) indeed achieves overall better results than a quadratic choice of kernel size (top right). Only for Condorcet and Independence, the quadratic choice is slightly better for some rules.

One might wonder, if one could still leverage diagonal correlations by first reordering the rankings of the voters in a profile so that 'similar' rankings are next to each other, before feeding the profile into the CNN. With such a pre-processing of the input, a quadratic kernel could be used since diagonal comparisons now make sense in such a similarity reordered profile. A standard way to formalize this notion of similarity is via Kendall Tau distance (as defined in footnote 8). We consider two versions of reordering a given profile $\boldsymbol{P} = (P_1, \dots, P_n)$:

- *Global*: Compute, for $k = 2, \dots, n$, the Kendall Tau distance d_k between P_1 and P_k. Then reorder the profile starting with P_1 followed by the other rankings with ascending d_k. (In case of a tie, pick the ranking with minimal index first.)
- *Local*: The reordered profile $\boldsymbol{P'} = (P'_1, \dots, P'_n)$ is computed recursively. Start with $P'_1 := P_1$. Given P'_k, we determine P'_{k+1} as follows. Go through the rankings that have not been picked yet (i.e., $\{P_1, \dots, P_n\} \setminus \{P'_1, \dots, P'_k\}$) and compute their Kendall Tau distance to P'_k. Then P'_{k+1} is the ranking among these with the smallest Kendall Tau distance. (Again, tie-break via the indices.)

Figure 8 shows that adding either the global or the local version of Kendall Tau pre-processing overall does not reliably help the performance compared to our chosen setting (neither for our choice of kernel size nor for the quadratic choice). In some cases we do observe an improvement, as for example in the independence axiom satisfaction when learning the Copeland rule—however, this always comes with an additional loss, either in the accuracy or in the satisfaction of other axioms such as anonymity and neutrality.

A.3 Cross Validation

Finally, we corroborate our choice of hyperparameters by establishing their robust learning capabilities via cross validation in Table 3. For this, we IC-sample a fixed dataset of $100,000$ data points. We split the dataset into 10 folds (each of size $10,000$). Looping over $k = 0, \dots, 9$, we take fold k as the test set and train the model on the data in the other 9 folds for 8 epochs. We record the achieved accuracy and loss (both on the training and the test set). Table 3 shows that, for all architectures with their chosen hyperparameters, we always get a high accuracy with little variance. This corroborates the robust learning capabilities of our architectures.

B Experiment 1

We run the 'correct for the right reasons' experiment from Section 6.1 in more settings, reported in Figure 9 (part 1) and Figure 10 (part 2).

C Experiment 2

We add further results to the 'learning principles by example' experiment from Section 6.2. Figure 11 and 12 show different choices of architecture, rules, and distribution for the first version of the experiment. Figure 13 and 14 show different choices of architecture, rules, and distribution for the second version of the experiment.

D Experiment 3

We add further results on the 'rule synthesis guided by principles' experiment (Section 6.3).

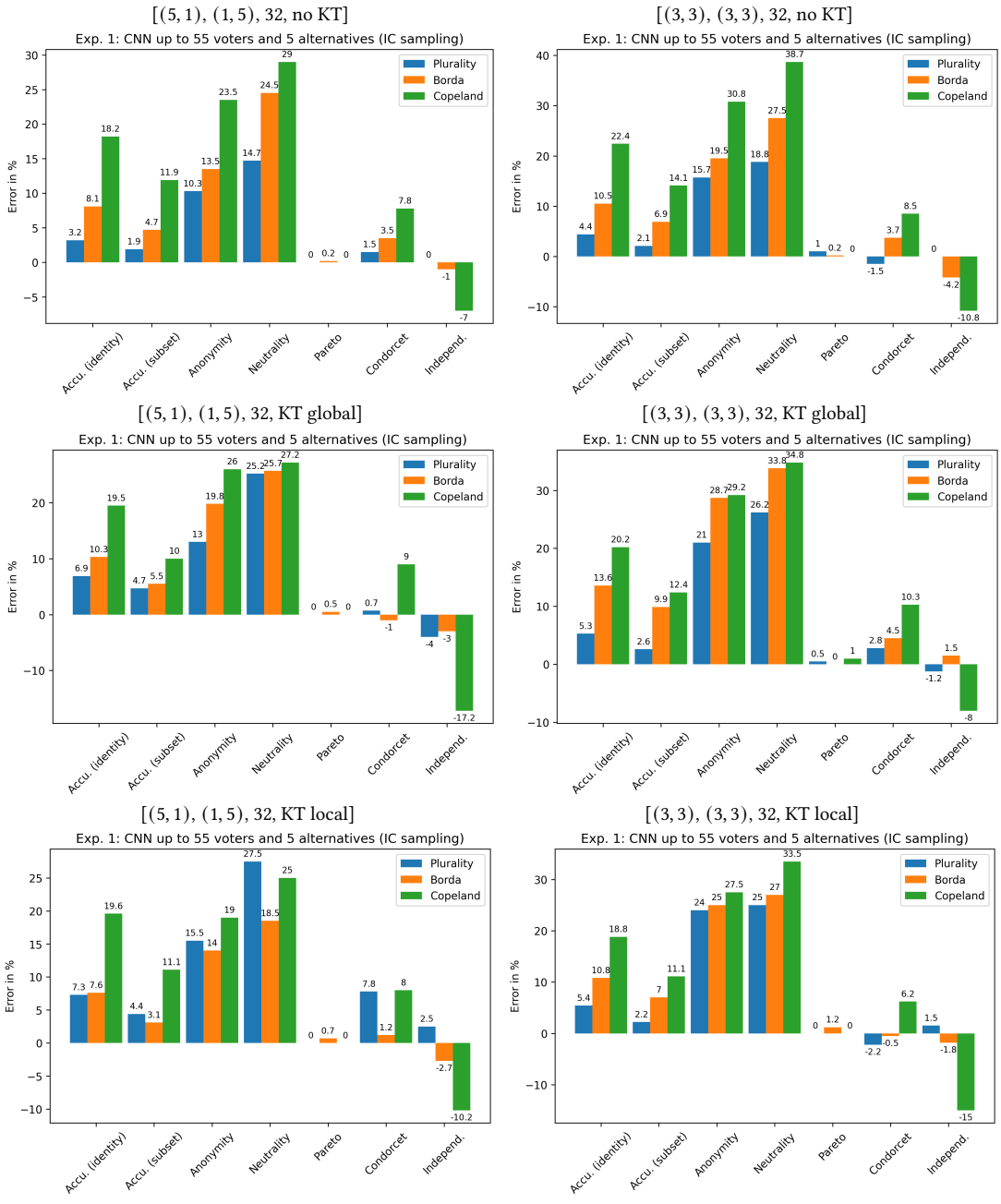

Fig. 8. Different hyperparameters of CNNs and their performance. The description [(5, 1), (1, 5), 32, no KT] means that, for this CNN, the first kernel has size (5, 1), the second kernel has size (1, 5), the number of channels is 32, and the input is not Kendall Tau preprocessed. Similarly for the other descriptions. (The top left plot is our standard CNN setting and repeated from Figure 9 for convenience.)

Table 3. Cross validation of the three architectures on a dataset with $100,000$ data points IC-sampled from the Plurality rule with up to 55 voters and 5 alternatives. Training is for 8 epochs with a batch size of 200 (hence $8 \times \frac{90,000}{200} = 3,600$ gradient steps).

MLP

Testing fold number	Train loss	Train accuracy (in %)	Test loss	Test accuracy (in %)
0	0.013	97.7	0.031	95.1
1	0.013	97.7	0.033	94.8
2	0.012	97.9	0.032	95.4
3	0.009	98.5	0.031	95.4
4	0.018	96.9	0.038	94.4
5	0.018	96.9	0.04	94.1
6	0.025	95.8	0.042	93.7
7	0.011	97.9	0.03	95.1
8	0.013	97.7	0.029	95.9
9	0.008	98.7	0.025	96
Avg.	0.014	97.6	0.033	95
Std. dev.	0.005	0.8	0.005	0.7

CNN

Testing fold number	Train loss	Train accuracy (in %)	Test loss	Test accuracy (in %)
0	0.021	96.5	0.023	96.1
1	0.015	97.2	0.019	96.8
2	0.009	98.7	0.011	98.3
3	0.012	98	0.015	97.3
4	0.016	97.3	0.019	96.9
5	0.01	98.5	0.011	98
6	0.009	98.4	0.012	98
7	0.02	96.3	0.02	96.4
8	0.014	97.3	0.019	96.6
9	0.024	95.8	0.029	95.4
Avg.	0.015	97.4	0.018	97
Std. dev.	0.005	0.9	0.005	0.9

WEC

Testing fold number	Train loss	Train accuracy (in %)	Test loss	Test accuracy (in %)
0	0.005	99.4	0.006	99.5
1	0.005	99.7	0.005	99.8
2	0.006	99.6	0.006	99.5
3	0.007	99.3	0.008	99.1
4	0.035	96.3	0.037	95.8
5	0.004	99.8	0.004	99.9
6	0.012	98.6	0.014	98.5
7	0.01	99	0.011	98.7
8	0.01	98.5	0.009	98.6
9	0.011	98.4	0.011	98.3
Avg.	0.01	98.8	0.011	98.8
Std. dev.	0.009	1	0.009	1.1

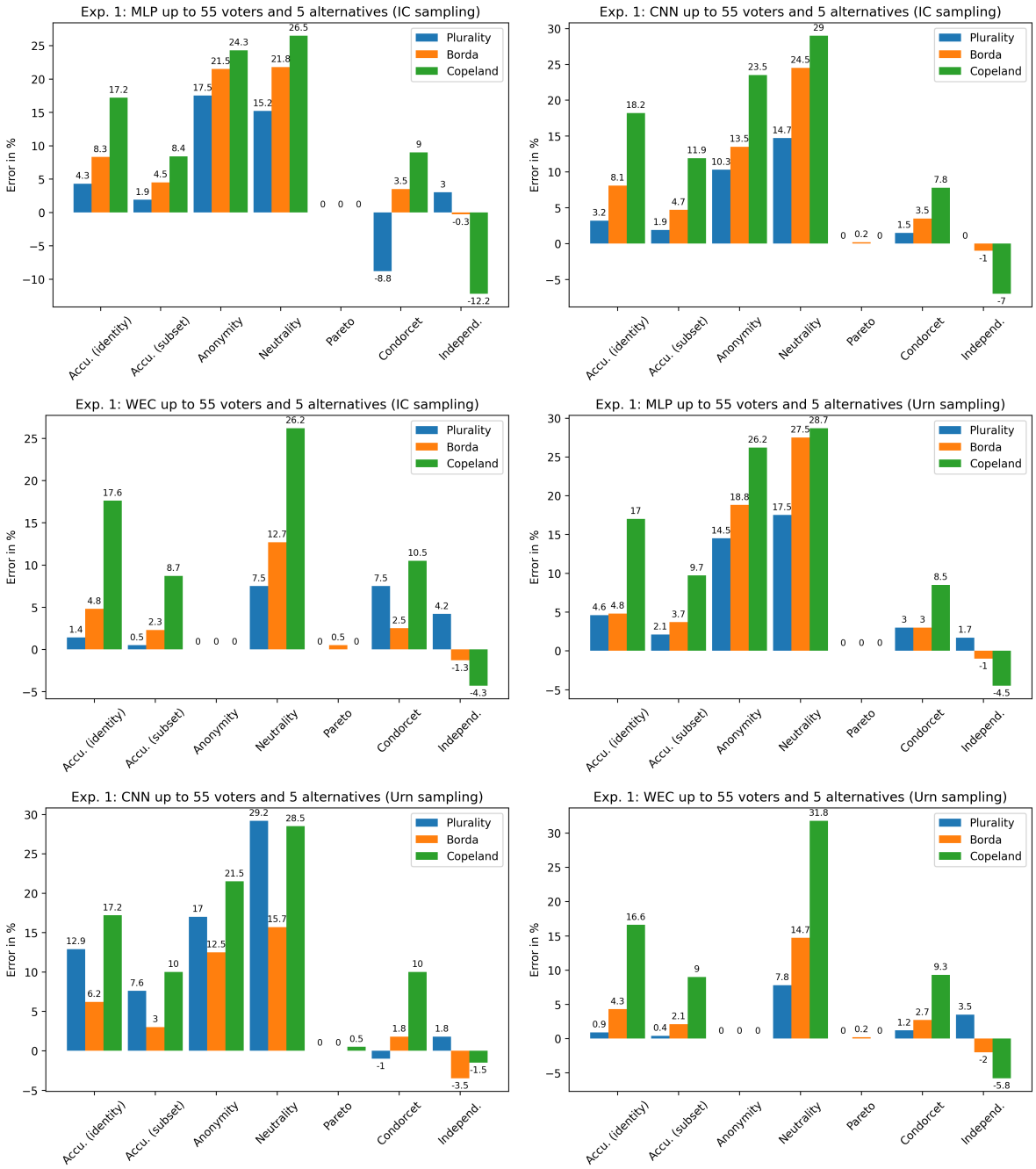

Fig. 9. Part 1 of more settings of experiment 1 (Section 6.1). Varying architectures, rules, and sampling, while comparing the errors in both accuracy and axiom satisfaction.

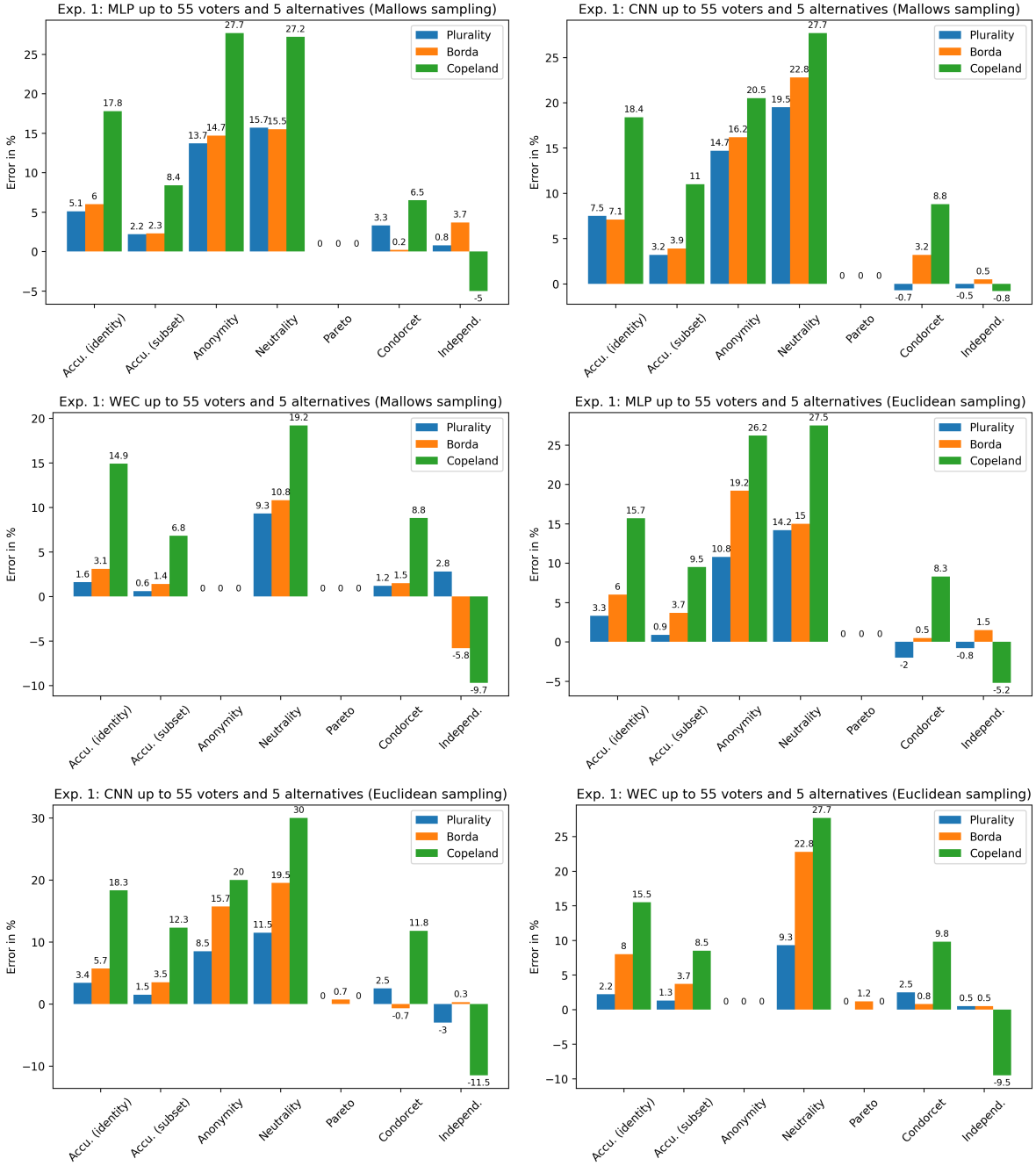

Fig. 10. Part 2 of more settings of experiment 1 (Section 6.1). Varying architectures, rules, and sampling, while comparing the errors in both accuracy and axiom satisfaction.

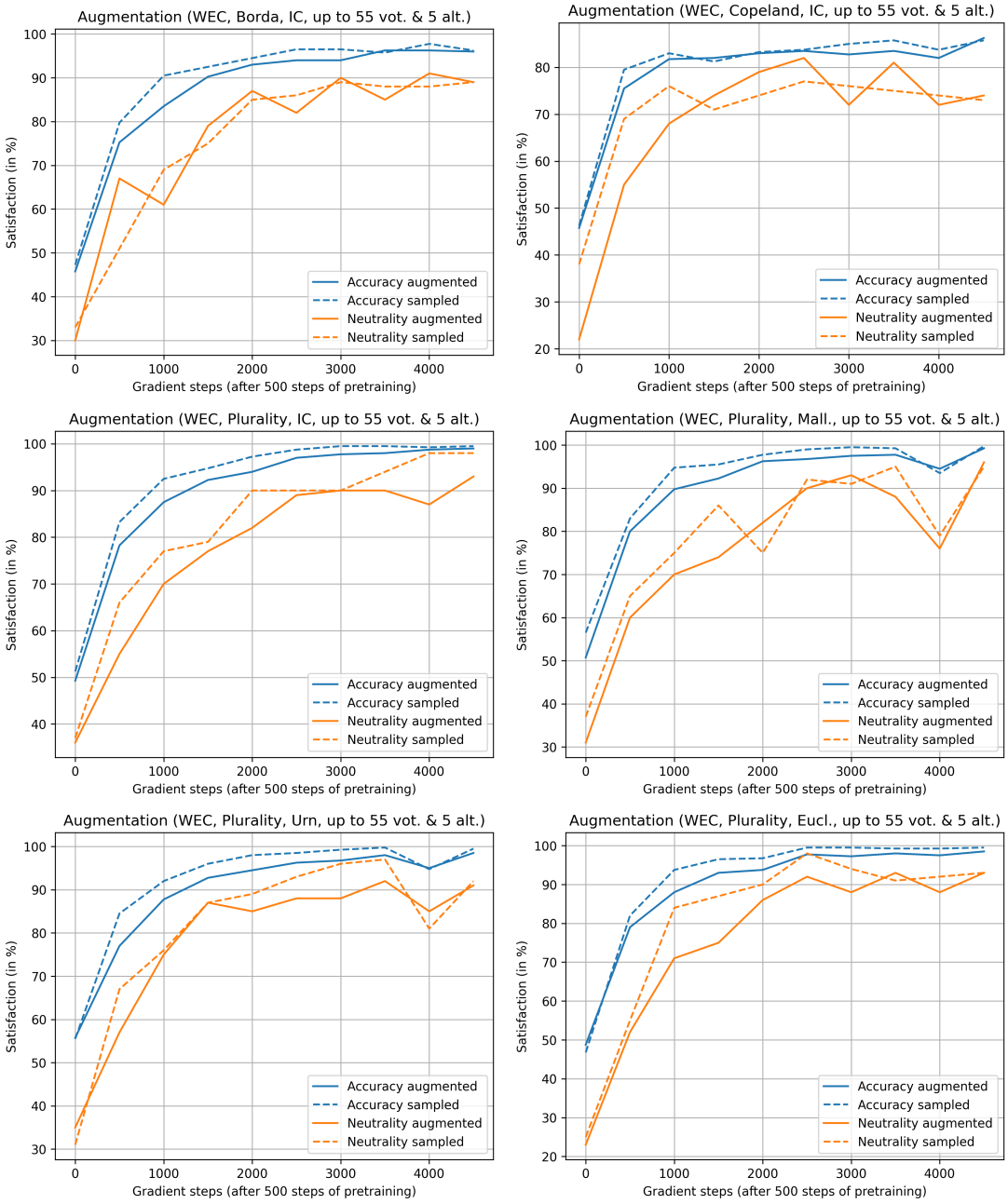

Fig. 11. The first version of experiment 2 (Section 6.2) with further architectures, rules, and distributions.

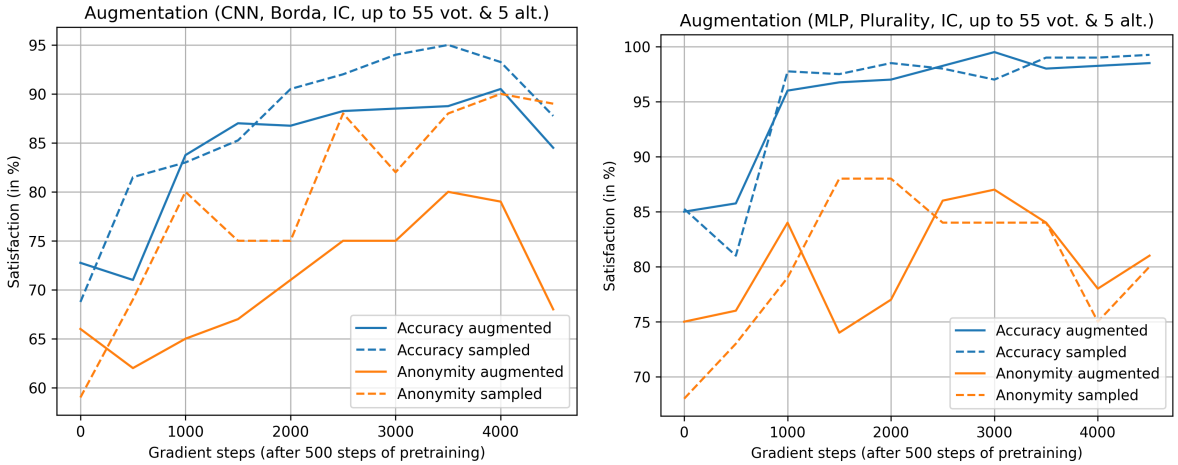

Fig. 12. The first version of experiment 2 (Section 6.2), but for the anonymity axiom (instead of neutrality). Since the WEC is anonymous by design, this can only be tested for MLP and CNN.

Table 4. IC sampling: The result of keeping on training an WEC model. Each round adds 20k gradient steps to the previous one. Round 1 is with a learning rate of 10^{-3}, round 2 with 10^{-4}, round 3 with $5 * 10^{-5}$, and round 4 the same but with added optimization of independence.

	Anon.	Neut.	Condorcet	Pareto	Indep.	Average
WEC n (NW, C, P, round 0)	100	100	97.5	100	46	88.7
WEC n (NW, C, P, round 1)	100	100	100	100	38.5	87.7
WEC n (NW, C, P, round 2)	100	100	100	100	34.8	87
WEC n (NW, C, P, I, round 3)	100	100	100	100	31.8	86.3

D.1 More on the Experiment with IC

Regarding difference-making profiles, Figure 6 in the main text shows a profile where the model *strongly disagrees* with its 5 closest voting rules, i.e., the model's winning set does not intersect the winning set of these rules. Figure 15 here shows a profile where the model *weakly disagrees* with all considered voting rules, i.e., the model's winning set is not identical with the winning set of these rules. (For the main text profile, the model also happens to weakly disagree with all considered rules.) We found these profiles by going through 10,000 IC-sampled profiles and picking the weakly or strongly disagreeing profile with the smallest number of voters.

Table 4 suggests that further optimization does not further improve axiom satisfaction.

Table 5 shows the statistical robustness of the axiom satisfaction achieved by the model.

Figure 16 shows the evolution of the semantic losses during optimization and the run times, showing that the WEC not only achieves the best results but also does so most quickly.

D.2 The Experiment with Mallows

Table 6 shows the result of the experiment from Section 6.3 but with Mallows sampling (instead of IC sampling), using the parameters discussed in Section 5.1. Table 7 shows how similar the best model is to its closest rules.

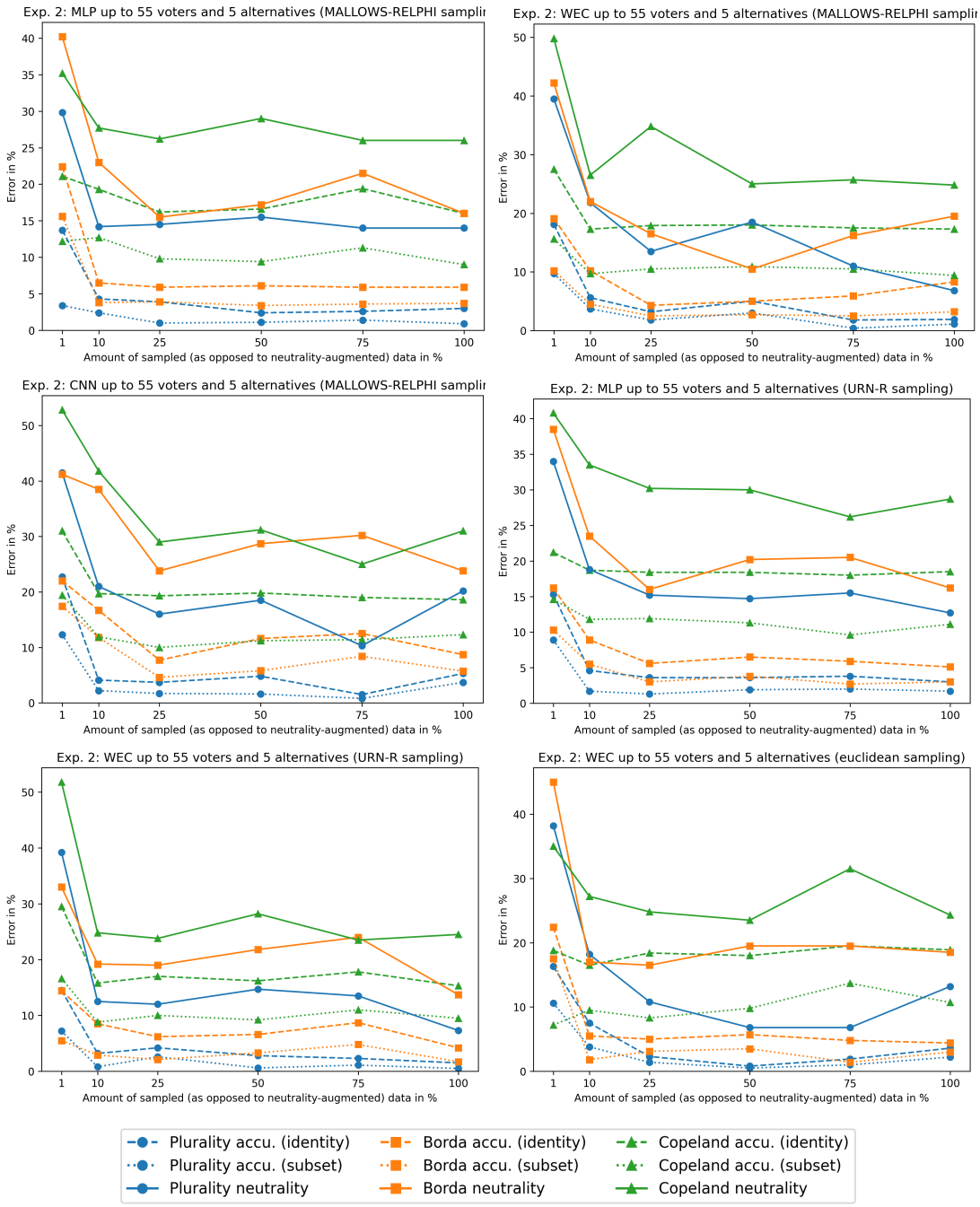

Fig. 13. The second version of experiment 2 (Section 6.2) with different distributions.

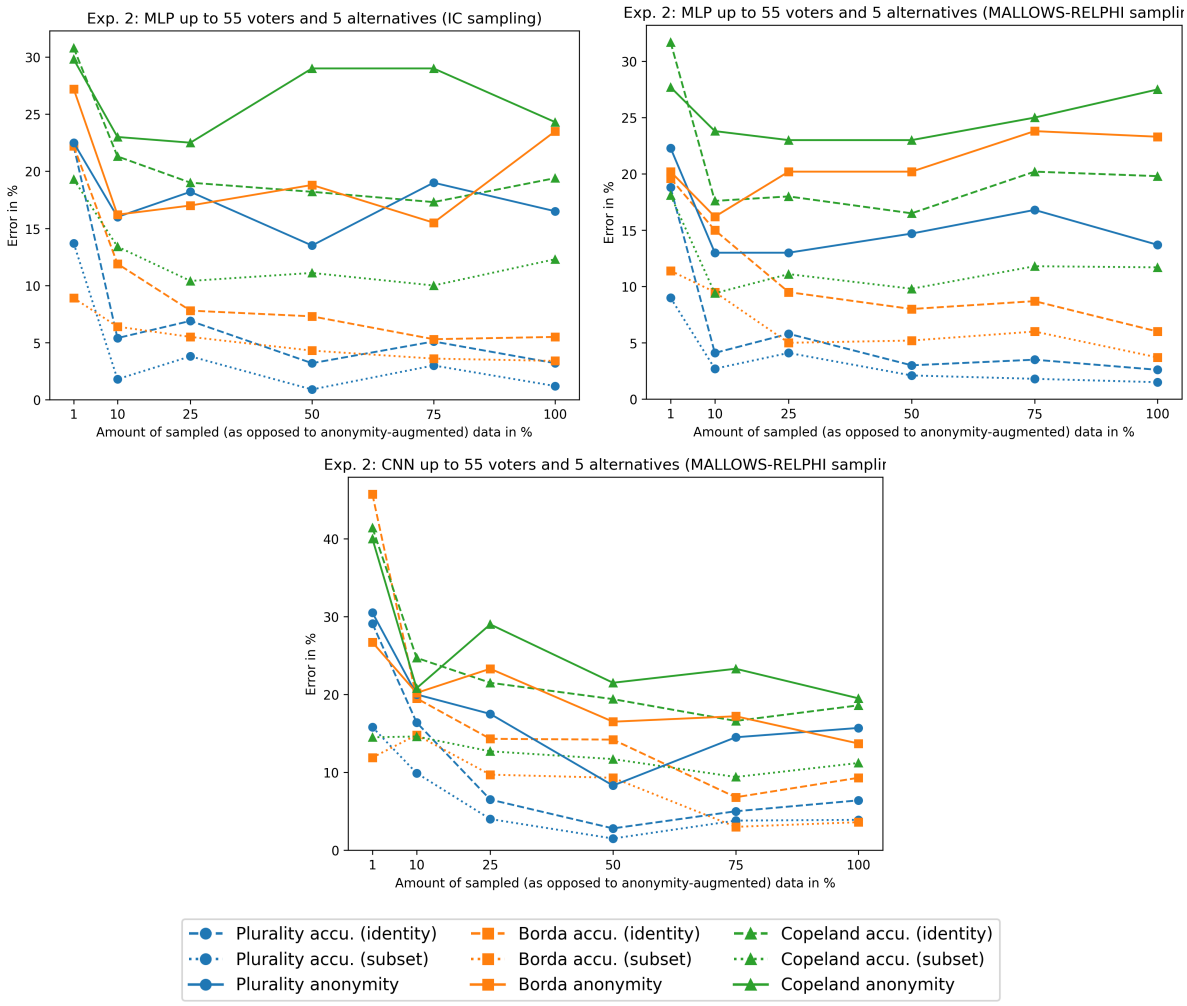

Fig. 14. The second version of experiment 2 (Section 6.2), but for the anonymity axiom (instead of neutrality). Since the WEC is anonymous by design, this can only be tested for MLP and CNN.

Figure 17 presents a profile where the model differs from all considered rules. Table 8 shows the statistical robustness of the axiom satisfaction achieved by the model.

D.3 The Experiment with Urn

Table 9 shows the result of the experiment from Section 6.3 but with Urn sampling (instead of IC sampling). Table 10 shows how similar the best model is to its closest rules. Figure 18 presents a profile where the model strongly differs (i.e., had non-intersecting winning sets) from its five closest rules. Table 11 shows the statistical robustness of the axiom satisfaction achieved by the model.

1	2	3	4	5	6	7
a	b	c	e	b	b	d
e	d	e	c	c	c	e
d	c	a	d	a	a	b
b	a	d	a	d	d	c
c	e	b	b	e	e	a

$\{b, c\}$	neutrality-averaged WEC, with sigmoids (rounded) a:.38, b:.54, c:.50, d:.39, e:.39
$\{b\}$	Plurality, Weak Nanson, Kemeny-Young
$\{c\}$	Borda, Copeland, Llull, Blacks, Coombs
$\{d\}$	Anti-Plurality, Baldwin
$\{e\}$	Instant Runoff TB
$\{b, d\}$	Stable Voting
$\{b, c, d\}$	Uncovered Set, Banks
$\{b, c, e\}$	PluralityWRunoff PUT
$\{a, b, c, d, e\}$	Top Cycle

Fig. 15. IC sampling: Profile where the 'WEC n' model weakly disagrees (i.e. non-identical winning sets) with existing voting rules.

Table 5. IC sampling: Take the average over 5 runs of checking the axiom satisfaction of the 'WEC n' model and its closest rules.

	Anon.	Neut.	Condorcet	Pareto	Indep.	Avg.
Blacks	100	100	100	100	36.04	87.2
Stable Voting	100	100	100	100	40.48	88.1
Borda	100	100	93.82	100	37.72	86.32
Weak Nanson	100	100	100	100	38.28	87.68
Copeland	100	100	100	100	28.54	85.72
WEC n (NW, C, P)	100	100	96.78	100	45.9	88.54

D.4 The Experiment with Euclidean

Table 12 shows the result of the experiment from Section 6.3 but with Euclidean sampling (instead of IC sampling). Table 13 shows how similar the best model is to its closest rules. Figure 19 presents a profile where the model differs from all considered rules. Table 14 shows the statistical robustness of the axiom satisfaction achieved by the model.

D.5 Ablation Study

Figure 20 displays an ablation study in the choice of axioms to optimize for. For the best performing model, i.e., the neutrality-averaged WEC, there remain three axioms that can be optimized for: Condorcet (C), Pareto (P), and independence (I). The 'No winner' loss (NW) is always needed to prevent the model from never outputting any winner. Which subset of the three axioms is the optimal choice for optimization? In the main text, we chose C and P. The figure shows that this choice indeed is the best one. Figure 21 shows, for each choice of axiom optimization, the evolution of the losses during training.

Table 6. Mallows sampling: Axiom satisfaction of different rules (top part of the table) and models (bottom part of the table). Otherwise like Table 1 from the main text.

	Anon.	Neut.	Condorcet	Pareto	Indep.	Average
Plurality	100	100	83.0	100	30.2	82.7
Borda	100	100	92.8	100	32.8	85.1
Anti-Plurality	100	100	76.5	100	26.2	80.5
Copeland	100	100	100	100	27.8	85.5
Llull	100	100	100	100	26.2	85.2
Uncovered Set	100	100	100	100	29.5	85.9
Top Cycle	100	100	100	100	25.2	85.0
Banks	100	100	100	100	25.2	85.0
Stable Voting	100	100	100	100	39.0	87.8
Blacks	100	100	100	100	33.8	86.8
Instant Runoff TB	100	100	96.8	100	29.0	85.2
PluralityWRunoff PUT	100	100	94.0	100	27.0	84.2
Coombs	100	100	95.5	100	30.2	85.2
Baldwin	100	100	100	100	39.2	87.9
Weak Nanson	100	100	100	100	33.8	86.8
Kemeny-Young	100	100	100	100	38.2	87.7
MLP p (NW, A, C, P, I)	78.8	76.0	94.0	100	38.8	77.5
MLP n (NW, A, C, P, I)	90.8	100	94.2	100	36.2	84.2
MLP na (NW, A, C, P, I)	92.5	89.5	92.5	100	33.5	81.6
CNN p (NW, A, C, P, I)	80.5	68.8	94.5	100	38.8	76.5
CNN n (NW, A, C, P, I)	91.5	100	95.0	100	42.2	85.8
CNN na (NW, A, C, P, I)	88.0	83.8	94.0	100	36.5	80.5
WEC p (NW, C, P)	100	65.5	91.8	100	37.8	79
WEC n (NW, C, P)	100	100	97.0	100	44.0	88.2

Table 7. Mallows sampling: Similarities between the rules. Computed on 10,000 sampled profiles. Otherwise like Table 2 from the main text.

Identity accuracy	WEC n	Stable Voting	Blacks	Borda	Weak Nanson	Copeland
WEC n	100	89.1	89.4	88.3	87.3	87.2
Stable Voting		100	95.61	91.04	93.47	92.16
Blacks			100	95.43	91.71	90.82
Borda				100	87.14	86.25
Weak Nanson					100	92.08
Copeland						100

Subset accuracy	WEC n	Stable Voting	Blacks	Borda	Weak Nanson	Copeland
WEC n	100	91.7	92.2	92.5	92.1	94.4
Stable Voting	95.8	100	97.09	94.4	97.78	99.49
Blacks	95.8	96.63	100	97.31	95.96	97.97
Borda	94.2	92.06	95.43	100	91.39	93.4
Weak Nanson	92.7	94.5	93.02	90.33	100	97.4
Copeland	91.3	92.23	91.6	88.91	94.17	100

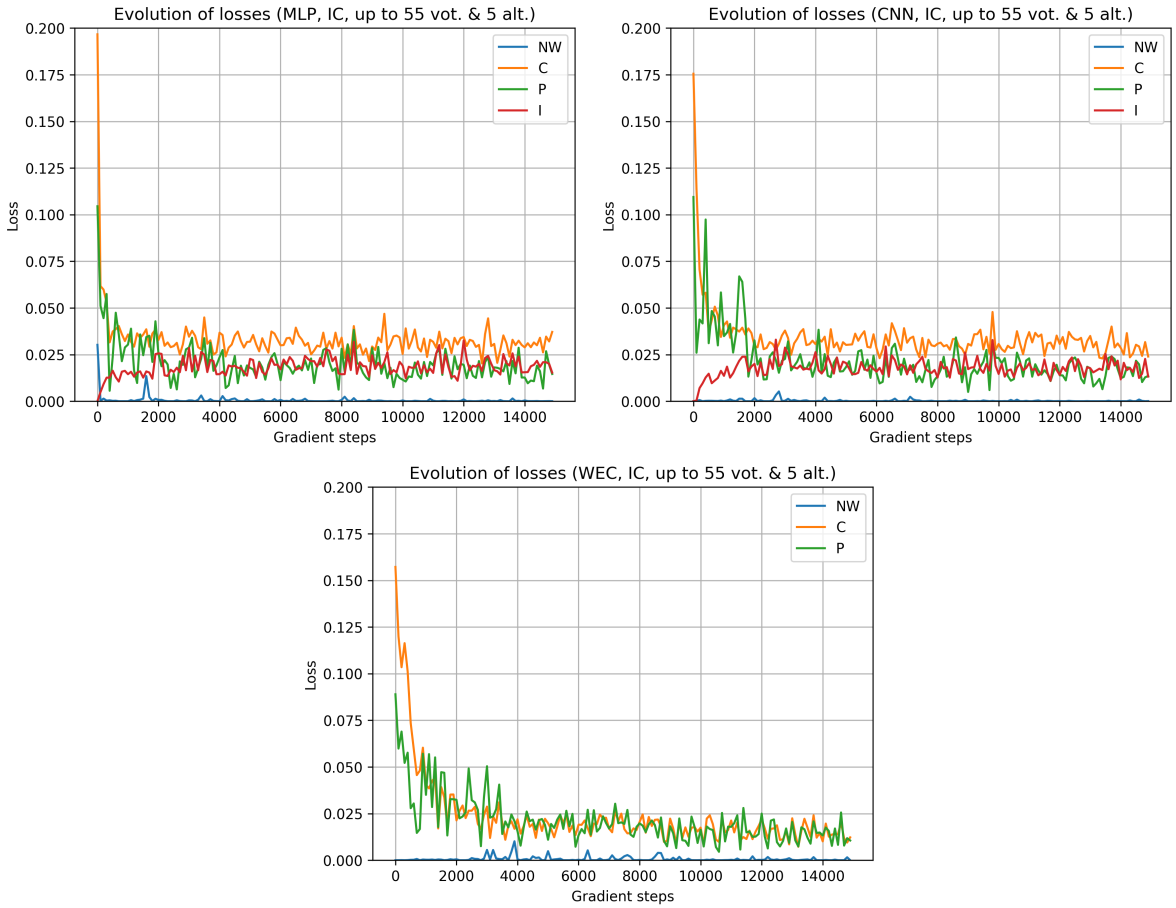

Fig. 16. The loss evolution during experiment 3. The run times of optimization were as follows: 5h 29min (MLP), 6h 08min (CNN), 2h 05min (WEC).

Table 8. Mallows sampling: Take the average over 5 runs of checking the axiom satisfaction of the 'WEC n' model and its closest rules.

	Anon.	Neut.	Condorcet	Pareto	Indep.	Avg.
Stable Voting	100	100	100	100	38.14	87.62
Blacks	100	100	100	100	34.04	86.84
Borda	100	100	94.16	100	35.02	85.84
Weak Nanson	100	100	100	100	37.18	87.46
Copeland	100	100	100	100	27.5	85.48
WEC n (NW, C, P)	100	100	94.92	100	41.72	87.34

Table 9. Urn sampling: Axiom satisfaction of different rules (top part of the table) and models (bottom part of the table). Otherwise like Table 1 from the main text.

	Anon.	Neut.	Condorcet	Pareto	Indep.	Avg.
Plurality	100	100	84.2	100	24.5	81.8
Borda	100	100	94.8	100	35	85.9
Anti-Plurality	100	100	76.8	100	25	80.3
Copeland	100	100	100	100	28.2	85.6
Llull	100	100	100	100	27.3	85.5
Uncovered Set	100	100	100	100	25.2	85
Top Cycle	100	100	100	100	27.8	85.5
Banks	100	100	100	100	27.5	85.5
Stable Voting	100	100	100	100	38	87.6
Blacks	100	100	100	100	34.2	86.9
Instant Runoff TB	100	100	97	100	28.2	85
PluralityWRunoff PUT	100	100	94.2	100	26.2	84.1
Coombs	100	100	96.5	100	28.5	85
Baldwin	100	100	100	100	39	87.8
Weak Nanson	100	100	100	100	38.5	87.7
Kemeny-Young	100	100	100	100	39.2	87.9
MLP p (NW, A, C, P, I)	79.8	74.2	92.8	100	34.8	76.3
MLP n (NW, A, C, P, I)	91	100	93.8	100	38.8	84.7
MLP na (NW, A, C, P, I)	91.8	91	93.5	100	35	82.2
CNN p (NW, A, C, P, I)	82.8	73.8	94	100	37.8	77.6
CNN n (NW, A, C, P, I)	90	100	94.5	100	34.5	83.8
CNN na (NW, A, C, P, I)	90	91.5	93.2	100	34.5	81.8
WEC p (NW, C, P)	100	75.2	93	100	37.8	81.2
WEC n (NW, C, P)	100	100	93.5	100	39	86.5

Table 10. Urn sampling: Similarities between the rules. Computed on 10,000 sampled profiles. Otherwise like Table 2 from the main text.

Identity accuracy	WEC n	Blacks	Stable Voting	Borda	Copeland	Weak Nanson
WEC n	100	88.9	88.5	88.2	86.5	86.3
Blacks		100	95.52	95.32	90.92	91.6
Stable Voting			100	90.84	92.17	93.61
Borda				100	86.24	86.92
Copeland					100	92.09
Weak Nanson						100

Subset accuracy	WEC n	Blacks	Stable Voting	Borda	Copeland	Weak Nanson
WEC n	100	92.3	91.4	93.2	94.3	91.3
Blacks	95.1	100	96.44	97.48	98.07	95.4
Stable Voting	94.9	97.03	100	94.51	99.47	97.48
Borda	93.9	95.32	91.76	100	93.39	90.72
Copeland	90.4	91.83	92.19	89.31	100	93.9
Weak Nanson	91.7	93.04	94.65	90.52	97.43	100

1	2	3	4
c	b	b	c
d	d	d	a
a	a	c	b
b	c	a	d

$\{b\}$	neutrality-averaged WEC, with sigmoids (rounded) a:.24, b:.51, c:.49, d:.32
$\{c\}$	Instant Runoff
$\{b, c\}$	Plurality, Borda, Copeland, Llull, Uncovered Set, Stable Voting, Blacks, PluralityWRunoff PUT, Baldwin, Weak Nanson, Kemeny-Young
$\{b, c, d\}$	Banks
$\{a, b, c, d\}$	Anti-Plurality, Top Cycle, Coombs

Fig. 17. Mallows sampling: Profile where the 'WEC n' model weakly disagrees (i.e. non-identical winning sets) with existing voting rules.

1	2	3	4	5
b	a	b	d	e
c	c	e	a	c
d	e	d	c	a
e	d	a	b	b
a	b	c	e	d

$\{b\}$	neutrality-averaged WEC, with sigmoids (rounded) a:.44, b:.50, c:.48, d:.40, e:.44
$\{c\}$	Blacks, Stable Voting, Borda, Copeland, Weak Nanson, Llull
$\{b\}$	Plurality, Instant Runoff TB
$\{a\}$	Baldwin
$\{a, b\}$	PluralityWRunoff PUT
$\{a, c\}$	Kemeny-Young
$\{a, c, e\}$	Uncovered Set, Banks
$\{a, b, c, d, e\}$	Anti-Plurality, Top Cycle, Coombs

Fig. 18. Urn sampling: Profile where the 'WEC n' model strongly disagrees (i.e. non-intersecting winning sets) with its 5 closest rules (among the remaining rules it only agrees with Plurality and Instant Runoff TB).

Table 11. Urn sampling: Take the average over 5 runs of checking the axiom satisfaction of the 'WEC n' model and its closest rules.

	Anon.	Neut.	Condorcet	Pareto	Indep.	Avg.
Blacks	100	100	100	100	34.38	86.9
Stable Voting	100	100	100	100	39.2	87.84
Borda	100	100	93.46	100	35.66	85.84
Copeland	100	100	100	100	26.9	85.38
Weak Nanson	100	100	100	100	37.82	87.54
WEC n (NW, C, P)	100	100	95.68	100	38.16	86.76

Table 12. Euclidean sampling: Axiom satisfaction of different rules (top part of the table) and models (bottom part of the table). Otherwise like Table 1 from the main text.

	Anon.	Neut.	Condorcet	Pareto	Indep.	Avg.
Plurality	100	100	79.8	100	25.5	81
Borda	100	100	93.8	100	37.2	86.2
Anti-Plurality	100	100	77.2	100	25	80.5
Copeland	100	100	100	100	29.5	85.9
Llull	100	100	100	100	29.8	86
Uncovered Set	100	100	100	100	26.2	85.2
Top Cycle	100	100	100	100	28.7	85.8
Banks	100	100	100	100	28	85.6
Stable Voting	100	100	100	100	39.5	87.9
Blacks	100	100	100	100	37	87.4
Instant Runoff TB	100	100	96.8	100	30	85.4
PluralityWRunoff PUT	100	100	94.8	100	26.5	84.2
Coombs	100	100	96	100	26.2	84.5
Baldwin	100	100	100	100	40.2	88
Weak Nanson	100	100	100	100	40	88
Kemeny-Young	100	100	100	100	36.5	87.3
MLP p (NW, A, C, P, I)	79.2	78	92.2	100	36	77.1
MLP n (NW, A, C, P, I)	87.2	100	93.8	100	36.2	83.5
MLP na (NW, A, C, P, I)	90.5	91.5	93.5	100	31.8	81.5
CNN p (NW, A, C, P, I)	83.2	74.5	93.2	100	35	77.2
CNN n (NW, A, C, P, I)	89.8	100	92	100	34.8	83.3
CNN na (NW, A, C, P, I)	86	90.2	92.8	100	30.5	79.9
WEC p (NW, C, P)	100	75	96.8	100	38.5	82
WEC n (NW, C, P)	100	100	97.8	100	42.2	88

Table 13. Euclidean sampling: Similarities between the rules. Computed on 10,000 sampled profiles. Otherwise like Table 2 from the main text.

Identity accuracy	WEC n	Stable Voting	Blacks	Weak Nanson	Copeland	Borda
WEC n	100	92.2	91.5	89.5	88.7	89.1
Stable Voting		100	95.65	93.22	92.24	91.05
Blacks			100	91.37	90.85	95.4
Weak Nanson				100	92.46	86.77
Copeland					100	86.25
Borda						100

Subset accuracy	WEC n	Stable Voting	Blacks	Weak Nanson	Copeland	Borda
WEC n	100	94.9	94.5	94.6	96.7	93.9
Stable Voting	95.6	100	97.19	97.5	99.58	94.57
Blacks	94.7	96.69	100	95.61	97.92	97.38
Weak Nanson	92	94.24	92.82	100	97.53	90.2
Copeland	90.8	92.3	91.67	94.39	100	89.05
Borda	92.1	92.09	95.4	91.01	93.32	100

1	2
d	b
a	e
c	a
b	c
e	d

$\{b\}$	neutrality-averaged WEC, with sigmoids (rounded) a:.34, b:.50, c:.13, d:.30, e:.13
$\{a\}$	Coombs
$\{d\}$	Instant Runoff TB
$\{a, b\}$	Weak Nanson
$\{b, d\}$	Plurality, PluralityWRunoff PUT
$\{a, b\}$	Borda, Copeland, Stable Voting, Blacks
$\{a, b, d\}$	Llull, Uncovered Set, Banks, Baldwin, Kemeny-Young
$\{a, b, c\}$	Anti-Plurality
$\{a, b, c, d, e\}$	Top Cycle

Fig. 19. Euclidean sampling: Profile where the 'WEC n' model weakly disagrees (i.e. non-identical winning sets) with existing voting rules.

Table 14. Euclidean sampling: Take the average over 5 runs of checking the axiom satisfaction of the 'WEC n' model and its closest rules.

	Anon.	Neut.	Condorcet	Pareto	Indep.	Avg.
Stable Voting	100	100	100	100	39.12	87.8
Blacks	100	100	100	100	34.98	86.98
Weak Nanson	100	100	100	100	39.32	87.86
Copeland	100	100	100	100	27.68	85.54
Borda	100	100	95.08	100	35.46	86.08
WEC n (NW, C, P)	100	100	97.74	100	45.16	88.58

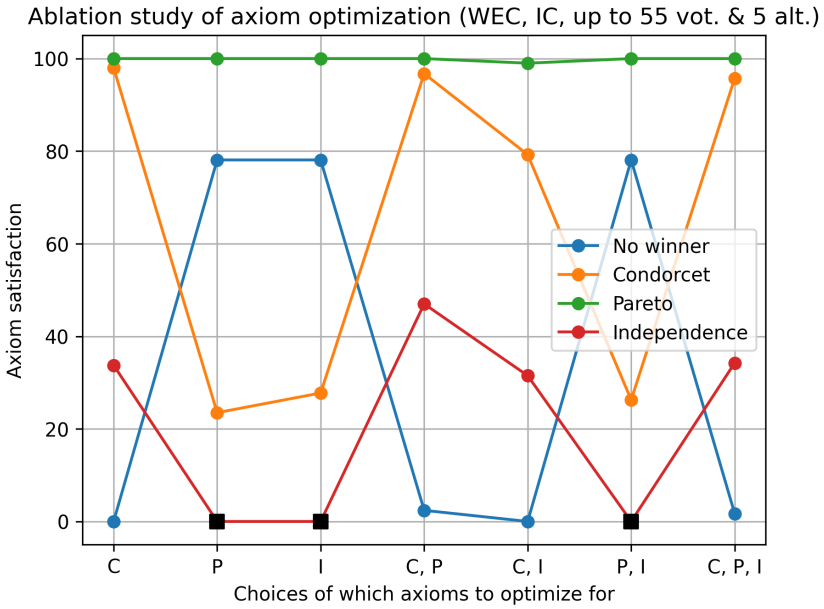

Fig. 20. Ablation study in axiom optimization with the neutrality-averaged WEC. For each possible (nonempty) choice of axioms to optimize for among Condorcet (C), Pareto (P), and independence (I), the achieved axiom satisfaction is shown. The 'No winner' loss (NW) is always optimized for. Its reported satisfaction is 0 if the model always outputs at least one winner. The axioms of anonymity and neutrality are not shown since they are satisfied by design. The black squares in the independence satisfaction indicate that the axiom was applicable on too few of the sampled test profiles to warrant an estimate. The best choice is C, P since it has the highest axiom satisfaction combined with a low 'No winner' satisfaction.

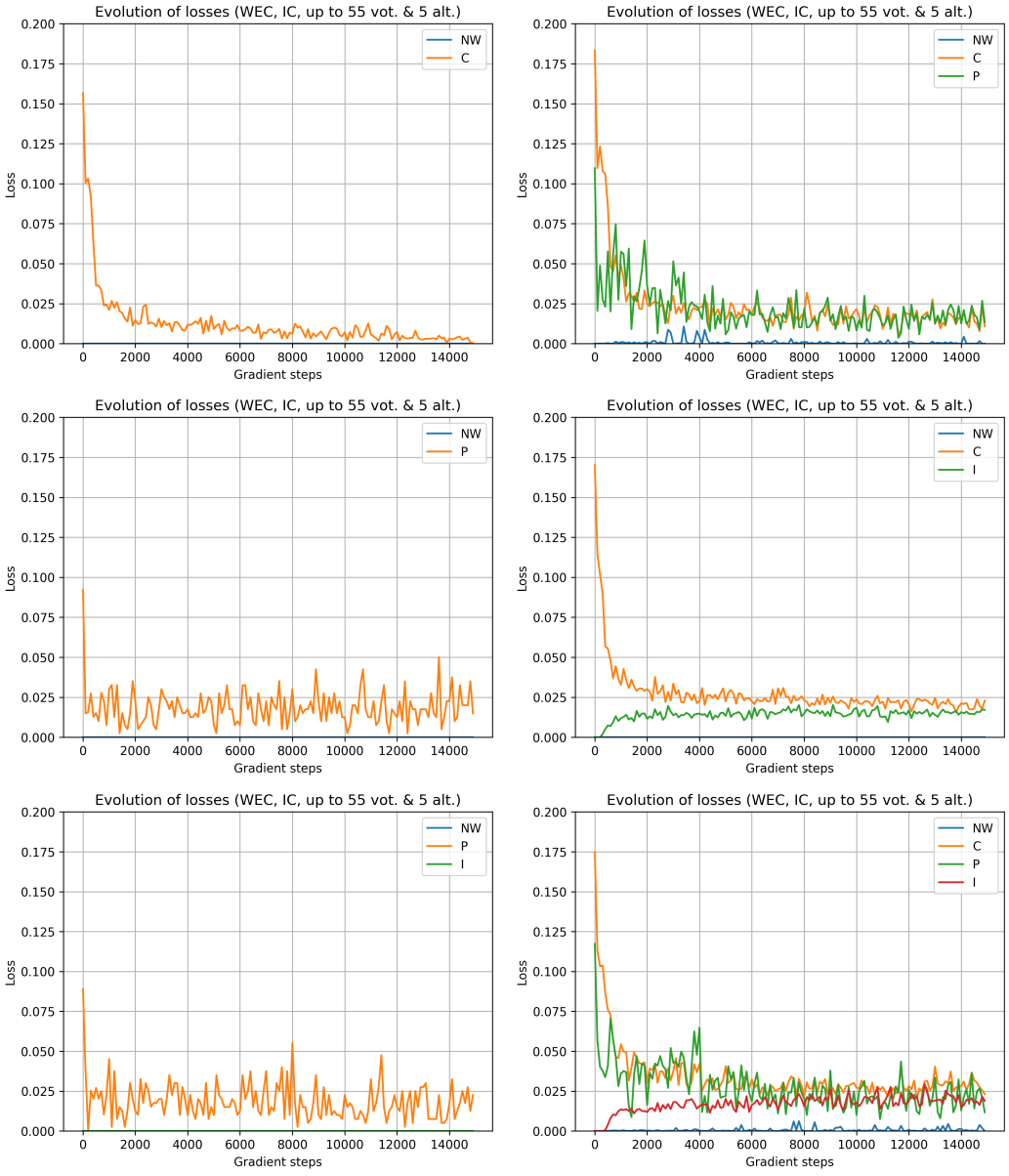

Fig. 21. The evolution of the losses during axiom optimization with the neutrality-averaged WEC, for each choice of which axioms to optimize among Condorcet (C), Pareto (P), and independence (I), with 'no winner' (NW) always being optimized for. The loss curves for the NW+I-optimization are not shown, since they are so close to 0 that they are indistinguishable from the x-axis. In the other two cases that did not yield good axiom satisfaction—i.e., P and P, I in Figure 20—, the loss evolution also shows no convergence.

Received 21 October 2024; revised 16 April 2025; accepted 14 June 2025